

**DETEKSI LARVA KEPITING FASE MEGALOPA  
MENGUNAKAN PENGOLAHAN CITRA**

**NOVI NURLAELA**

**D032172009**



**PROGRAM PASCASARJANA**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2019**

**DETEKSI LARVA KEPITING FASE MEGALOPA  
MENGUNAKAN PENGOLAHAN CITRA**

Tesis

Sebagai Salah Satu Syarat Untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun Dan Diajukan Oleh

**NOVI NURLAELA**

**Kepada**

**SEKOLAH PASCASARJANA**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2019**

# TESIS

## DETEKSI LARVA KEPITING FASE MEGALOPA MENGGUNAKAN PENGOLAHAN CITRA

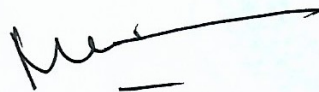
Disusun dan diajukan oleh

**NOVI NURLAELA**

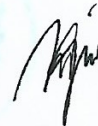
**Nomor Pokok D032172009**

telah dipertahankan di depan Panitia Ujian Tesis  
pada tanggal 16 Desember 2019  
dan dinyatakan telah memenuhi syarat

Menyetujui  
Komisi Penasehat,



**Muhammad Niswar, ST.MIT. Ph.D**  
Ketua



**Dr. Ir. Ingrid Nurtanio, MT**  
Sekretaris

Ketua Program Studi S2  
Teknik Elektro,



**Prof. Dr. Eng. Syafaruddin, ST., M.Eng.**  
NIP. 197405301999031003

Dekan Fakultas Teknik  
Universitas Hasanuddin,



**Prof. Dr. Ir. Muhammad Arsyad Thaha, MT.**  
NIP. 196012311986091001

## PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan dibawah ini:

Nama : Novi Nurlaela

Nomor Pokok : D032172009

Program Studi : Teknik Elektro

Program Studi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 16 Desember 2019

Yang menyatakan,

Novi Nurlaela

## KATA PENGANTAR

Alhamdulillah, Puji syukur penulis panjatkan kehadirat Allah SWT, atas nikmat dan kasih sayang-Nya yang telah memberikan kesehatan dan kekuatan untuk menyelesaikan tesis dengan judul “**Deteksi Larva Kepiting Fase Megalopa Menggunakan Pengolahan Citra**”. Tak lupa shalawat dan salam kepada Nabi Muhammad Rasulullah SAW panutan dan penuntun kepada akhlak yang mulia.

Pada kesempatan ini penulis menyampaikan rasa hormat dan menghaturkan terimakasih sebesar-besarnya kepada:

1. Bapak Muhammad Niswar, ST.MIT,Ph.D dan Ibu Dr.Ingrid Nurtanio, M.T atas bimbingan, arahan dan waktu yang telah diluangkan kepada penulis untuk berdiskusi selama menjadi dosen pembimbing.
2. Bapak Dr. Muhammad Anshar, ST.M.Sc, Bapak Dr. Ikhlas Kitta, ST.MT, Bapak Dr.Zulkifli Tahir, ST, M.Sc yang telah memberikan masukan dan saran pada saat seminar proposal dan seminar hasil tesis.
3. Seluruh Dosen program Pascasarjana Program Studi Teknik Informatika yang telah memberikan arahan untuk mendalami Teknik Informatika.
4. Bapak Prof.Dr.Irwan, M.Pd dan Bapak Yopi Sopian,ST.M.Pd selaku pimpinan di kantor LPPPTK KPTK yang telah memberikan izin dan dukungan selama penulis menempuh pendidikan Pascasarjana.
5. Ayahanda dan Ibunda serta saudara saya atas segala dukungan dan doa Nya

6. Suami saya Dwi Bagus Fitriyanto atas segala motivasi, perhatian, kesabaran dan doanya. Serta putra putri tercinta ananda Deandra Aditnya Zidni Dana dan Mutia Alisha Nandita yang selalu jadi penyejuk hati.
7. Rekan rekan mahasiswa Teknik Informatika 2017 teruntuk Yulia Ekawati, Magfirah Dinsyah, Nurfitri yang telah menjadi sahabat dalam susah maupun senang dalam menyelesaikan penelitian ini.
8. Kepada semua pihak yang telah membantu yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa tesis ini tentu masih belum sempurna. Dengan keterbatasan pengalaman, ilmu maupun pustaka yang ditinjau. Penulis mengharapkan kritik dan saran sebagai masukan bagi penulis untuk penelitian kedepannya serta mengharapkan semoga tulisan ini bisa memberikan manfaat kepada seluruh pihak.

Makasar, 16 Desember 2019

Penulis

## ABSTRAK

**Novi Nurlaela.** Deteksi larva kepiting fase megalopa menggunakan pengolahan citra (dibimbing oleh Muhammad Niswar dan Ingrid Nurtanio)

Kepiting Perenang Biru memiliki nilai ekonomi tinggi sebagai komoditas ekspor. Salah satu masalah yang terjadi dalam budaya kepiting adalah kanibalisme selama fase pengembangan larva. Larva kepiting fase megalopa cenderung memakan larva kepiting fase zoea karena pemberian makan yang tidak memadai. Untuk menghindari kanibalisme dalam fase pengembangan larva kepiting, petani perlu memeriksa perkembangan larva secara teratur dan memisahkan larva fase kepiting megalopa dari larva fase kepiting zoea. Pemeriksaan dilakukan dengan melihat langsung ke bak pemeliharaan, dan hal tersebut memakan waktu dan sulit karena perubahan fase dalam larva kepiting tidak dapat diprediksi. Penelitian ini berfokus pada proses deteksi larva kepiting fase megalopa menggunakan metode convolutional neural network (CNN). Penelitian ini menggunakan 210 gambar, dan hasil deteksi mendapatkan akurasi rata-rata 87.7%.

Kata kunci: CNN, deteksi, larva kepiting, pengolahan citra.

## ABSTRACT

**Novi Nurlaela.** The Detection of Megalopa Phase Crab Using Image Processing (supervised by Muhammad Niswar and Ingrid Nurtanio).

Blue Swimmer Crab has high economic value as an export commodity. One of the problems that occur in crab culture is cannibalism during the larval development phase. Megalopa phase crab larvae tend to eat zoea phase crab larvae due to inadequate feeding. To avoid cannibalism in the crab larvae development phase, farmers need to check the development of larvae regularly and separate the megalopa phase crab larvae from zoea phase crab larvae. Checking is done by looking directly at the rearing tank, which is time-consuming and difficult because the phase changes in the crab larvae are unpredictable. This research focuses on the detection process of the megalopa phase crab larvae using a convolutional neural network (CNN) method. This research uses 210 images, and detection results get shown an average accuracy of 87.7%..

Keywords : CNN, Crab Larvae, Detection, Image Processing, Megalopa



**DAFTAR ISI**

HALAMAN SAMPUL.....	i
PERNYATAAN KEASLIAN TESIS.....	iv
KATA PENGANTAR .....	v
ABSTRAK .....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL .....	xiv
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah .....	4
C. Tujuan Penelitian .....	4
D. Manfaat Penelitian .....	4
E. Batasan Masalah .....	5
F. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	8
A.Landasan Teori .....	8
B.Penelitian Terkait.....	28

C.State Of The Art .....	31
BAB III METODE PENELITIAN.....	37
A.Tahap Penelitian .....	37
B.Waktu Dan Lokasi Penelitian .....	38
C.Jenis Penelitian.....	39
D.Perancangan Sistem.....	39
E.Sumber Data .....	59
F.Instrumen Penelitian.....	60
G. Tolak Ukur Akurasi Sistem.....	61
BAB IV HASIL DAN PEMBAHASAN.....	62
A. Akuisisi Data .....	62
B.Analisa Sistem Deteksi Larva Kepiting Fase Megalopa .....	63
C.Analisis Kerja Sistem .....	69
BAB V KESIMPULAN DAN SARAN .....	79
A.Kesimpulan .....	79
B.Saran.....	79
DAFTAR PUSTAKA.....	81
DAFTAR LAMPIRAN .....	84

**DAFTAR GAMBAR**

Gambar 1. Larva Fase Zoea 1 .....	9
Gambar 2. Larva Fase Zoea 2 .....	9
Gambar 3. Larva Fase Zoea 3 .....	10
Gambar 4. Larva Fase Zoea 4 .....	10
Gambar 5. Larva Fase Megalopa .....	11
Gambar 6. SVM berusaha menemukan hyperplane terbaik .....	15
Gambar 7. Arsitektur MLP sederhana .....	18
Gambar 8. Pembelajaran pada CNN .....	19
Gambar 9. Fitur ekstraksi layer .....	20
Gambar 10. Lapisan yang terdapat pada CNN .....	21
Gambar 11. contoh dari max pooling .....	22
Gambar 12. Langkah menghitung CNN .....	24
Gambar 13. Gambar dengan wilayah pembatas .....	25
Gambar 14. Hasil deteksi menggunakan YOLO .....	25
Gambar 15. Arsitektur Algoritma YOLO .....	27
Gambar 16. Kerangka pikir penelitian .....	36
Gambar 17. Tahapan penelitian .....	37
Gambar 18. Rancangan Sistem Deteksi Larva Fase Megalopa .....	39

Gambar 19. Code Pengambilan Gambar di Raspberry Pi .....	42
Gambar 20. Code Transfer data .....	43
Gambar 20. Tahapan Deteksi Larva menggunakan algoritma svm .....	44
Gambar 22. Implementasi Konvolusi .....	47
Gambar 23. Hasil Pengolahan implementasi konvolusi .....	47
Gambar 24. Proses tresholding .....	48
Gambar 25. Ekstraksi fitur .....	49
Gambar 26. Grafik svm .....	50
Gambar 27. Hasil klasifikasi menggunakan svm.....	51
Gambar 28. Tahapan Penggunaan Algoritma YOLOv3.....	53
Gambar 29. Gambar yang akan dilatih .....	54
Gambar 30. Prose Labeling menggunakan Yolo_Mark .....	54
Gambar 31. Hasil Labelling .....	55
Gambar 32. Pengaturan file konfigurasi.....	58
Gambar 33. Output Hasil Deteksi .....	59
Gambar 34. Posisi Kamera pada Bak larva .....	63
Gambar 35. Gambar uji .....	68
Gambar 36. Hasil deteksi.....	68
Gambar 37. Hasil deteksi menggunakan .....	68

Gambar 38. Pengaturan koneksi .....	69
Gambar 39. Pengaturan koneksi raspberi pi.....	70
Gambar 40. Pengujian koneksi raspberry pi dengan server .....	71
Gambar 41. Pengaktifan layanan gammu.....	71
Gambar 42. Pengecekan port gammu .....	72
Gambar 43. Pengaturan settingan port pada gammurc dan smsdrc.....	72
Gambar 44. Pengaturan gammu pada file gammurc dan smsrc.....	73
Gambar 45. Pengaktifan layanan database .....	74
Gambar 46. Pengecekan alamat ip raspberry pi.....	74
Gambar 47. Pengaturan alamat capture .....	75
Gambar 48. Pengaturan alamat output.....	75
Gambar 49. Lokasi penyimpanan file capture dan output.....	76
Gambar 50. Pengaturan durasi waktu capture image .....	76
Gambar 51. Hasil capture image .....	77
Gambar 52. Proses deteksi hasil capture imgae larva.....	77
Gambar 53. Penyimpanan hasil capture dan gambar hasil deteksi .....	78
Gambar 54. Tabel hasil pengiriman notifikasi .....	78
Gambar 55. Tampilan notifikasi .....	78

**DAFTAR TABEL**

Tabel 1. State of the Art .....	32
Tabel 2. Hasil pengujian sistem .....	63
Tabel 3. Proses uji dengan gambar megalopa dan bukan megalopa .....	65
Tabel 4. Gambar uji non megalopa .....	66

# BAB I

## PENDAHULUAN

### A. Latar Belakang

“Rajungan” atau *Blue Swimmer Crab* merupakan salah satu komoditas perikanan yang memiliki nilai ekspor penting bagi Indonesia. Kepiting rajungan memiliki distribusi yang luas keberbagai negara, kepiting rajungan memijah sepanjang tahun, telur kepiting rajungan mencapai 1 juta per induk tetapi kondisi dialam dengan kehidupan burayak yang panjang menyebabkan kehilangan yang besar menjadi alasan pembudidayaan kepiting(Juwana, 1997). Dalam mendorong produksi rajungan, kepiting, dan juga lobster sehingga produksinya tidak tergantung di alam, salah satu upaya yang dilakukan Kementerian Kelautan dan Perikanan (KKP) melakukan pembudidayaan. BPBAP Takalar merupakan salah satu UPT KKP yang menjadi pelaksana dalam pembudidayaan tersebut. Hasil budidaya tersebut memiliki tujuan salah satunya melakukan restocking (penebaran kembali) di alam(“Bernilai Ekonomi Tinggi, KKP Benihkan Rajungan secara Massal di Takalar | MaritimNews.com,” n.d.).

Permasalahan yang terjadi pada budidaya salah satu diantaranya kanibalisme, dimana larva fase megalopa cenderung makan sesamanya ketika makanan yang diberikan tidak mencukupi(Juwana, 1997). Sehingga

memungkinkan dapat terjadinya kanibalisme apabila ukuran kepiting yang dipelihara tidak seragam lagi terutama setelah hari ke sepuluh selama masa pemeliharaan(Djunaedi, 2009). Kanibalisme dapat ditekan dengan salah satu cara yaitu pemberian tempat berlindung baik berupa *shelter* maupun *substrat* dasar yang cocok, *grading* dan pengurangan kepadatan larva selama pemeliharaan(Zmora et al., 2005).

Dalam usaha menekan kanibalisme pada larva kepiting tersebut petani perlu melakukan pengecekan perkembangan fase larva kepiting. Pengecekan tersebut dilakukan secara konvensional dengan melihat langsung kekolam larva selama pemeliharaan dan hal tersebut memerlukan banyak waktu. Untuk lebih mengefisienkan waktu dalam proses pengecekan fase larva kepiting tersebut pemanfaatan teknologi otomatisasi berbasis pengolahan citra diperlukan untuk mengetahui fase larva kepiting. Dalam pengolahan citra, citra yang ada diolah sedemikian rupa sehingga citra tersebut lebih mudah untuk diproses lebih lanjut(Arymurthy, 1992).

Teknologi pengolahan citra digital mulai diterapkan di bidang budidaya perikanan. Dedy Harto dkk mengembangkan perangkat lunak untuk mendeteksi penyakit *white spot* pada udang, penyakit *white Spot Syndrome Virus* dapat menyebabkan kematian udang hingga 100% menggunakan teknologi citra digital dan jaringan syaraf tiruan (JST)akurasi pengujian mencapai 93.33%(Harto and Sarosa, 2012). Rosa Andrie dkk dengan implementasi pengolahan citra digital mengembangkan alat yang mampu menghitung benih ikan. Sistem dirancang secara *realtime* menggunakan



bahasa pemrograman *Python* menggunakan metode pengolahan citra *Thresholding*, *Morphology*, serta *labelling*. Pada pengujian 40 benih ikan tersebut akurasi mencapai 99%(A et al., 2017). Sheryl May D.Lainez dkk melakukan penelitian penghitungan benih ikan secara otomatis menggunakan teknik image processing dan melakukan investigasi tentang efektifitas Convolutional Neural Network (CNN) dalam mendeteksi ikan dan akurasi perhitungannya. Hasil percobaannya menunjukkan bahwa CNN dapat digunakan dalam produksi pembenihan, tingkat deteksi dan akurasi perhitungannya yang tinggi(Lainez and Gonzales, 2019a).

Pengolahan citra dalam beberapa tahun terakhir mengalami perkembangan yang cukup luar biasa begitu halnya dengan deep learning, deep learning yang merupakan bagian dari machine learning dengan algoritma yang terinspirasi oleh struktur dan fungsi otak yang biasa disebut konsep jaringan syaraf tiruan (JST). YOLO "You Only Look Once" merupakan salah satu deep learning yang diperkenalkan dengan pendekatan baru untuk deteksi objek. YOLO memiliki kemampuan pada base model yang bisa memproses 45 frame per second secara real-time dan pada versi yang lebih kecil bisa memproses 155 frame per second secara real-time(Redmon et al., 2016).

Pada penelitian sebelumnya melalui sejumlah besar percobaan, mengeksplorasi deteksi gambar sampel kecil yang cepat dan akurat metode di bawah kerangka belajar mendalam Yolo. Hasil menunjukkan bahwa untuk set data sampel kecil optik, pra-pelatihan aktif data sampel besar yang tidak terkait untuk mendapatkan bobot, dan kemudian menggunakan pelatihan dan

prediksi jaringan Yolo adalah metode yang sangat efektif. Hasilnya menunjukkan bahwa YOLO tidak hanya akurat, tetapi juga cepat, dan kinerja pendeteksian dapat ditingkatkan melalui peningkatan sampel(Li et al., 2018). Metode ini sejalan dengan sistem yang diusulkan untuk mendeteksi larva kepiting pada fase megalopa dimana objek larva memiliki ukuran yang relatif kecil. Otomatisasi dengan memanfaatkan teknologi pengolahan citra diperlukan dalam rangka meningkatkan efektifitas pada pengecekan larva kepiting.

### **B. Rumusan Masalah**

Bagaimana membuat sistem yang dapat mendeteksi larva kepiting fase megalopa berdasarkan ukuran larva menggunakan pengolahan citra?

### **C. Tujuan Penelitian**

Adapun tujuan penelitian yang ingin dicapai yaitu untuk mendeteksi larva kepiting fase megalopa berdasarkan ukuran objek larva menggunakan pengolahan citra.

### **D. Manfaat Penelitian**

Manfaat dari penelitian deteksi larva kepiting fase megalopa ini yaitu agar dapat memberikan peringatan dini ke petambak ketika larva kepiting telah

memasuki fase megalopa sehingga petambak dapat segera memisahkan antara larva besar dan larva kecil. Hal ini sangat berguna bagi petambak untuk menekan jumlah larva kecil yang mati akibat kanibalisme pada fase megalopa, sehingga jumlah tingkat kelulus hidupan larva kepiting meningkat guna pemenuhan kebutuhan akan bibit dan jumlah panen kedepannya dapat lebih optimal.

### **E. Batasan Masalah**

Adapun batasan masalah penelitian larva kepiting fase megalopa ini:

1. Input berupa file citra dari larva kepiting jenis *Blue Swimmer*
2. Proses pengambilan citra larva kepiting dilakukan di kolam pembibitan dilakukan saat kondisi cukup terang pencahayaan dalam ruangan.
3. Pengambilan gambar tidak mencakup seluruh kolam tetapi hanya pada batas area yang sudah ditentukan

### **F. Sistematika Penulisan**

Adapun sistematika penulisan pada penelitian pembuatan sistem deteksi larva kepiting fase megalopa adalah:

#### **Bab I Pendahuluan**

Bab I berisi penjelasan tentang latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian dengan objek larva kepiting *Blue Swimmer* untuk mendeteksi fase megalopa.

## **Bab II Landasan Teori dan Kerangka Pemikiran**

Bab II berisi penjelasan tentang tinjauan pustaka yang merupakan penjelasan tentang hasil-hasil penelitian lainnya yang berkaitan dengan penelitian yang dilakukan tentang sumber acuan terbaru dari pustaka primer seperti buku, artikel, jurnal, dan tulisan asli lainnya untuk mengetahui penelitian yang relevan dengan usulan penulis terkait deteksi larva kepiting fase megalopa. Dalam bab ini juga diuraikan tentang kerangka pikir untuk memecahkan masalah yang sedang diteliti.

## **Bab III Metodologi Penelitian**

Bab III ini merupakan penjelasan tentang tahapan penelitian, bagaimana pengembangan dan penerapan sistem dengan metode pengolahan image processing yang diusulkan penulis untuk mendeteksi larva kepiting fase, rancangan sistem usulan penulis diuraikan proses validasi hasil dan metode usulan penulis serta analisa data pada hasil penerapan sistem yang diusulkan penulis.

## **Bab IV Hasil dan Pembahasan**

Pada bab IV ini menjelaskan tentang hasil dan pembahasan penelitian serta implikasi dari penelitian yang dilakukan. Hasil merupakan suatu penjelasan tentang data kuantitatif yang dikumpulkan sesuai dengan metodologi yang telah ditetapkan. Pembahasan merupakan suatu penjelasan tentang

pengolahan data dan interpretasinya, baik dalam bentuk deskriptif ataupun penarikan inferensinya. Implikasi penelitian merupakan suatu penjelasan tentang tindak lanjut penelitian yang terkait aspek sistem, maupun aspek penelitian lanjutan.

## **BAB V Kesimpulan dan Saran**

Pada bab V ini berisi ringkasan temuan, rangkuman kesimpulan dan saran. Kesimpulan merupakan pernyataan secara general atau spesifik yang berisi hal-hal penting dan menjadi temuan penelitian yang bersumber pada hasil dan pembahasan. Saran merupakan pernyataan atau rekomendasi peneliti yang berisi hal-hal penting sebagaimana yang telah di sampaikan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A.Landasan Teori**

##### **1) Larva Kepiting**

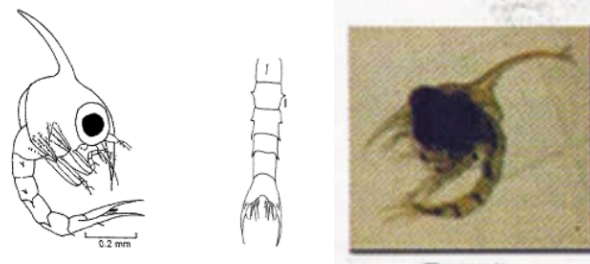
Rajungan dan kepiting merupakan Crustacea dari familia Portunidae, dan hidup di laut terbuka mulai dari pinggir pantai hingga kedalaman sekitar 30meter keduanya merupakan komoditi perikanan yang bagus, hewan ini juga memiliki tempat hidup/habitat yang beranekaragam. Kepiting blue swimmer atau rajungan dapat hidup pada habitat yang bermacam-macam, antara lain daerah pantai berpasir, daerah muara dan daerah pertambakan biasa yang bersubstrat lumpur(Darsono, 1997).

Daur hidup kepiting meliputi telur, larva (zoea dan megalopa), post larva atau juvenil, anakan dan dewasa. Perkembangan embrio dalam telur mengalami 9 fase (Jose, n.d.). Larva yang baru ditetaskan (tahap zoea) bentuknya lebih mirip udang dari pada kepiting. Di kepala terdapat semacam tanduk yang memanjang, matanya besar dan di ujung kaki-kakinya terdapat rambut-rambut. Tahap zoea ini juga terdiri dari 4 tingkat untuk kemudian berubah ke tahap megalopa dengan bentuk yang lain lagi. Larva kepiting berenang dan terbawa arus serta hidup sebagai plankton. Beberapa hasil penelitian menyebutkan bahwa larva kepiting hanya mengkonsumsi

fitoplankton beberapa saat setelah menetas dan segera setelah itu lebih cenderung memilih zooplankton sebagai makanannya.

### 1. Larva Fase Zoea 1

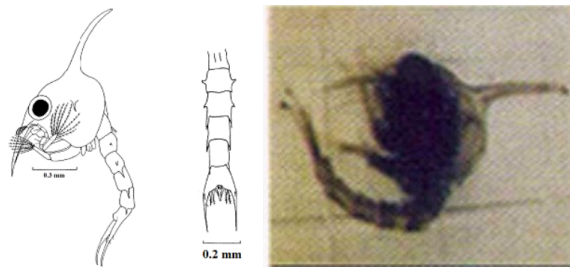
Pada fase zoea 1 variasi ukuran panjang karapas antara 0.44-0.54 mm dan panjang abdomen-telson antara 1.07 - 1.23 mm.



Gambar 1. Larva Fase Zoea 1

### 2. Larva Fase Zoea 2

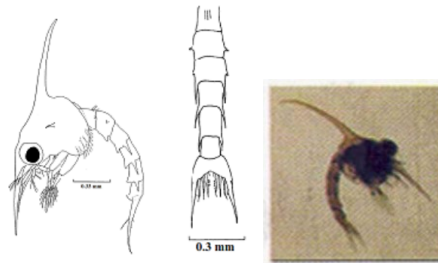
Pada fase zoea 2 variasi ukuran panjang karapas antara 0.72-0.77 mm dan panjang abdomen sampai telson antara 1.07 - 1.23 mm.



Gambar 2. Larva Fase Zoea 2

### 3. Larva Fase Zoea 3

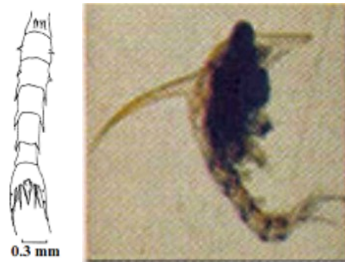
Pada fase zoea 3 variasi ukuran panjang karapas antara 0.79-0.87 mm dan panjang abdomen sampai telson antara 2.02-2.21 mm



Gambar 3. Larva Fase Zoea 3

#### 4. Larva Fase Zoea 4

Pada fase zoea 3 variasi ukuran panjang karapas antara 0.98-1.06 mm dan panjang abdomen sampai telson antara 2.61-3.03 mm

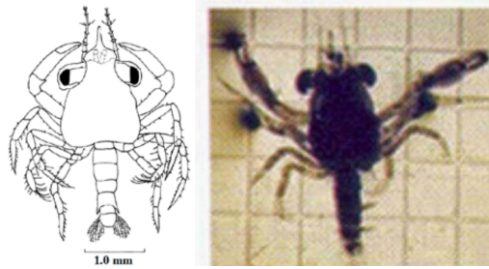


Gambar 4. Larva Fase Zoea 4

#### 5. Larva Fase Megalopa

Panjang karapas (termasuk rostrum) 1,69-1,81 mm, lebar 1,16-1,31 mm. Perut enam tersegmentasi, dengan telepon datar dorso-ventrally. Perut panjangnya (termasuk telson) 1.311.35 mm. Total panjang termasuk mimbar 3.0-3.2 mm.





Gambar 5. Larva Fase Megalopa

## 2) Citra

Citra (image) istilah lain untuk gambar— sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (a picture is more than a thousand words). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (scanner), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat:

1. optik berupa foto,
2. analog berupa sinyal video seperti gambar pada monitor televisi,
3. digital yang dapat langsung disimpan pada suatu pita magnetik.

Agar dapat diolah dengan dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut digitalisasi. Citra yang dihasilkan inilah yang disebut citra digital (digital image). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi · lebar (atau lebar · panjang). Citra digital yang tingginya N, lebarnya M, dan memiliki L derajat keabuan dapat dianggap sebagai fungsi:

$$f(x, y) \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases}$$

Citra digital yang berukuran N · M lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut:

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Masing-masing elemen pada citra digital (berarti elemen matriks) disebut image element, picture element atau pixel atau pel. Jadi, citra yang berukuran N · M

mempunyai NM buah pixel. Sebagai contoh, misalkan sebuah berukuran 256 x 256 pixel dan direpresentasikan secara numerik dengan matriks yang terdiri dari 256 buah baris (di-indeks dari 0 sampai 255) dan 256 buah kolom (di-indeks dari 0 sampai 255) seperti contoh berikut:

$$\begin{bmatrix} 0 & 134 & 145 & \dots & \dots & 231 \\ 0 & 167 & 201 & \dots & \dots & 197 \\ 220 & 187 & 189 & \dots & \dots & 120 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 221 & 219 & 210 & \dots & \dots & 156 \end{bmatrix}$$

Pixel pertama pada koordinat (0, 0) mempunyai nilai intensitas 0 yang berarti warna pixel tersebut hitam, pixel kedua pada koordinat (0, 1) mempunyai intensitas 134 yang berarti warnanya antara hitam dan putih, dan seterusnya.

Proses digitalisasi citra ada dua macam:

1. Digitalisasi spasial (x, y), sering disebut sebagai penerokan (sampling).
2. Digitalisasi intensitas  $f(x, y)$ , sering disebut sebagai kuantisasi.

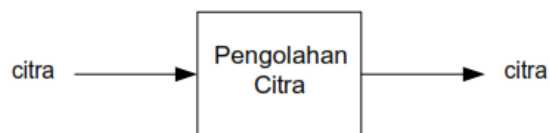
### 3) Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik.

Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila:

1. perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra,
2. elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur,
3. sebagian citra perlu digabung dengan bagian citra yang lain.

**Pengolahan Citra** bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini jug adalah pemampatan citra (*image compression*).



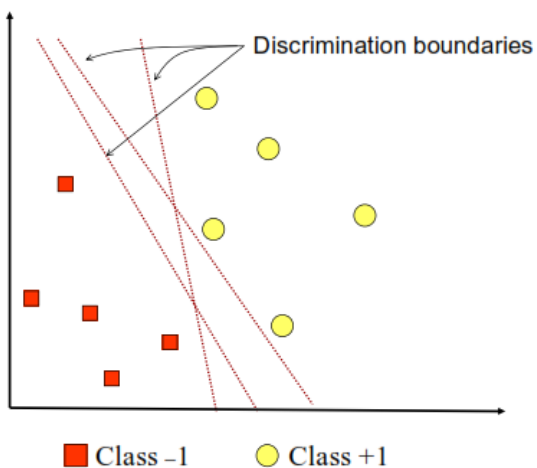
Citra digital di presentasikan dengan matriks. Operasi pada citra digital dasarnya adalah memanipulasi elemen-elemen matriks. Elemen matriks yang dimanipulasi dapat berupa elemen tunggal (sebuah pixel)(Arymurthy, 1992).

#### 4) Support Vector Machine (SVM)

SVM merupakan metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimalization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah class pada input space.

Berbeda dengan strategi neural network yang berusaha mencari hyperplane pemisah antar class, SVM berusaha menemukan hyperplane yang terbaik pada input space. Prinsip dasar SVM adalah linear classifier, dan selanjutnya dikembangkan agar dapat bekerja pada problem non-linear, dengan memasukkan konsep kernel trick pada ruang kerja berdimensi tinggi.

Konsep svm dapat dijelaskan secara sederhana sebagai salah satu usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space. Pada gambar di bawah ini memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class :+1(lingkaran kuning) dan -1(kotak merah). Problem klasifikasi dapat di terjemahkan dengan usaha menemukan garis (hyperplane) yang memisahkan antara kedua kelompok tersebut.



Gambar 6. SVM berusaha menemukan hyperplane terbaik

Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM.

Data yang tersedia dinotasikan sebagai  $\vec{x}_i \in \mathcal{R}^d$  sedangkan label masing-masing di notasikan  $y_i \in \{-1, +1\}$  untuk  $i=1, 2, \dots, l$  yang mana  $l$  adalah banyaknya data. Diasumsikan kedua class -1 dan +1 dapat terpisah secara sempurna oleh hyperplane berdimensi  $d$  yang di definisikan

$$\vec{w} \cdot \vec{x} + b = 0 \quad (1)$$

Pattern  $\vec{x}_i$  yang termasuk class -1 (sample negatif) dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \quad (2)$$

Sedangkan pattern  $\vec{x}_i$  yang termasuk class +1 (sampel positif)

$$\vec{w} \cdot \vec{x}_i + b \geq +1 \quad (3)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antar hyperplane dan titik terdekatnya, yaitu  $1/\|\vec{w}\|$ . Hal ini dapat di rumuskan sebagai *Qadratic Programming(QP) problem* yaitu mencari titik minimum

$$\min_{\vec{w}} \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (4)$$

Dengan memperhatikan Constrain persamaan

$$y_i (\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, \quad \forall i \quad (5)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi diantaranya Lagrange Multipliers

$$L(\bar{w}, b, \alpha) = \frac{1}{2} \|\bar{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i ((\bar{x}_i \cdot \bar{w} + b) - 1))$$

$$(i = 1, 2, \dots, l)$$
(6)

$\alpha_i$  adalah Lagrange Multipliers yang bernilai nol atau positif (Nilai optimal dari persamaan (6) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$  dan  $b$  dan memaksimalkan problem yang hanya mengandung saja  $\alpha_i$  sebagaimana persamaan berikut

Maximize :

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$
(7)

Subject to:

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0$$
(8)

Dari hasil dari perhitungan ini diperoleh yang kebanyakan bernilai positif. Data yang berkorelasi dengan yang positif inilah yang disebut sebagai support vector.

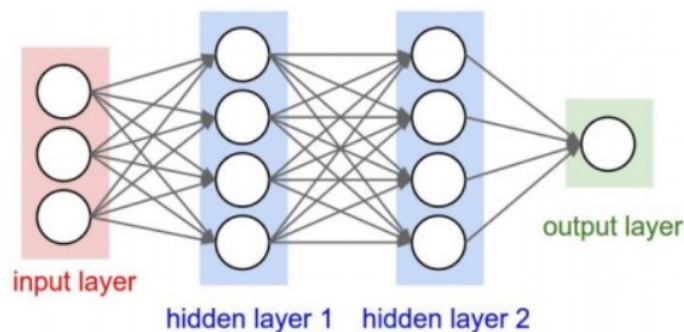
Secara prinsip SVM adalah linear classifier. Pattern recognition dilakukan dengan mentransformasikan data pada input space ke ruang yang berdimensi lebih tinggi dan optimisasi dilakukan pada ruang vektor yang baru tersebut.

Hal ini membedakan SVM dari solusi patters recognition pada umumnya yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi input space. Menerapkan strategi *Structural Risk Minimalization*(SRM). Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi 2 class.

### 5) Convolutional Neural Network (CNN)

Convolutional Nerual Nertork adalah salah satu metode machine learning dari pengembang Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk salah satu Deep Nerula Network karena dalamnya tingkat jaringan dan banyak di implementasikan dalam data citra. CNN memiliki dua metode yaitu klasifikasi menggunakan feedforward dan tahap pembelajaran menggunakan backpropagation.

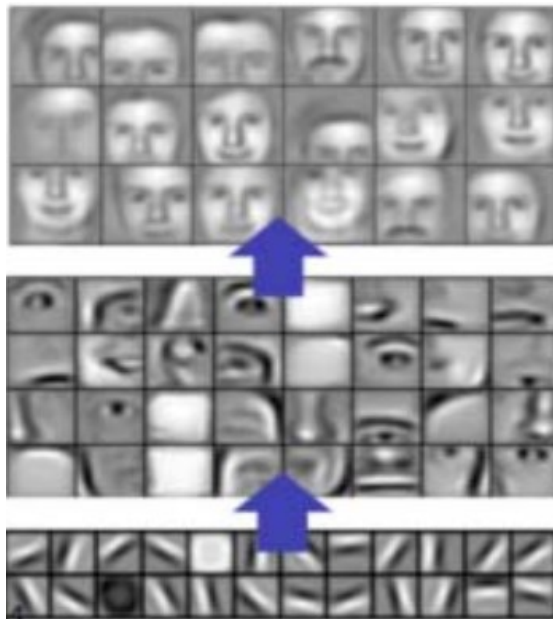
Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron di presentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi



Gambar 7. Arsitektur MLP sederhana



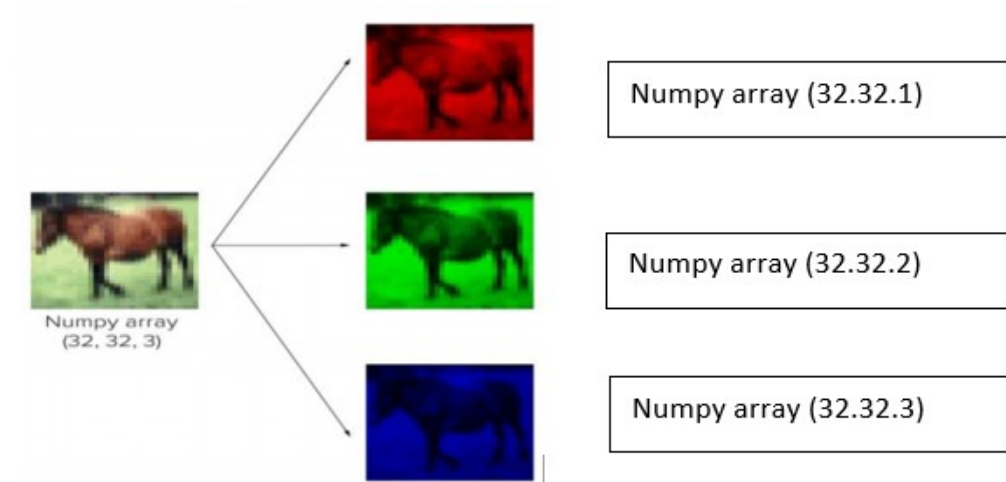
CNN memiliki terobosan selama dekade terakhir di berbagai bidang yang terkait pengenalan pola, pengolahan citra dan pengenalan suara. CNN memiliki keunggulan dalam hal jumlah parameter yang lebih sedikit serta mampu menyelesaikan tugas-tugas yang kompleks yang tidak dimiliki jaringan saraf tiruan klasik. Aspek penting lain dari CNN adalah fitur abstrak saat input merambat ke lapisan yang lebih dalam. Misalnya dalam klasifikasi gambar, pada lapisan pertama mendeteksi tepi, kemudian lapisan kedua mendeteksi yang lebih sederhana dan fitur selanjutnya yang lebih tinggi hal tersebut dapat dilihat pada contoh gambar 8 berikut ini



Gambar 8. Pembelajaran pada CNN

Convolutional layer adalah neuron yang disusun sedemikian rupa sehingga membentuk sebuah filter dengan ukuran lebar dan tinggi dalam pixel. Sebagai contoh, layer pertama pada Feature Extraction Layer biasanya adalah Convolution Layer dengan ukuran  $5 \times 5 \times 3$ . Panjang 5 pixels, tinggi 5 pixels dan

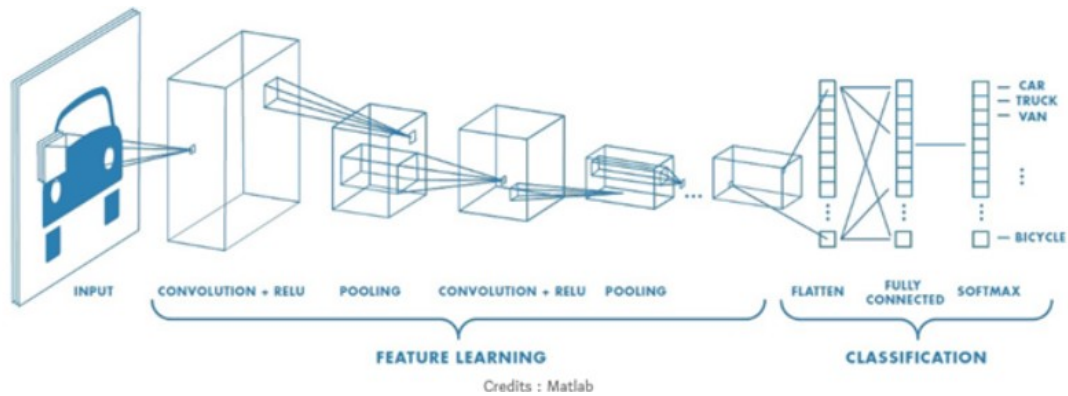
tebal/jumlah 3 buah sesuai dengan channel dari image tersebut. Gambar 7 adalah RGB (Red, Green, Blue) image berukuran 32x32x3.



Gambar 9. Fitur ekstraksi layer

Ketiga filter ini akan bergeser dengan jumlah pixel yang telah ditentukan keseluruhan bagian dari gambar. Operasi dot antara input dan nilai dari filter dilakukan pada setiap pergeseran sehingga menghasilkan sebuah output atau biasa disebut sebagai activation map atau feature map.

CNN membentuk neuron-neuronnya kedalam tiga dimensi(panjang, lebar dan tinggi) dalam sebuah lapisan. **(Albawi et al., n.d.)**.



Gambar 10. Lapisan yang terdapat pada CNN

Lapisan pada CNN terbagi menjadi :

### 1. Feature Learning

Lapisan pada feature learning berguna untuk mentranslasikan suatu input menjadi fitur berdasarkan ciri dari input tersebut yang membentuk angka-angka dalam vektor. Lapisan ekstraksi fitur ini terdiri dari Convolution layer dan Pooling layer

#### a) Convolutional layer

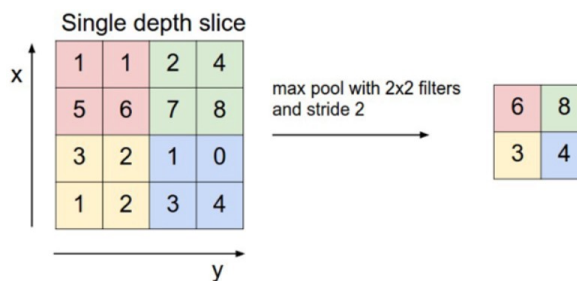
Menghitung output dari neuron yang terhubung ke daerah lokal dalam input, masing-masing produk titik antara bobot dan wilayah kecil yang terhubung kedalam volume input

#### b) Rectified Linear Unit (ReLU)

Menghilangkan vanishing gradient dengan cara menerapkan fungsi aktivasi elemen sebagai  $f(x)=\max(0,x)$  aktivasi elemen akan dilakukan saat berada di ambang batas 0.

### c) Pooling Layer

Ditempatkan setelah Convolution Layer. Pada prinsipnya Pooling Layer merupakan sebuah filter, yang akan bergeser pada seluruh area feature map, dengan ukuran dan stride tertentu. Teknik Pooling yang sering digunakan adalah Max Pooling dan Average Pooling. Pooling layer digunakan untuk mengurangi dimensi dari feature map (downsampling), sehingga komputasi dapat dilakukan lebih cepat karena parameter yang harus di-update semakin sedikit dan mengatasi overfitting.



Gambar 11. contoh dari max pooling

## 2. Classification

Lapisan ini berguna untuk mengklasifikasikan tiap neuron yang telah di ekstrakasi fitur pada sebelumnya.

### a) Flatten

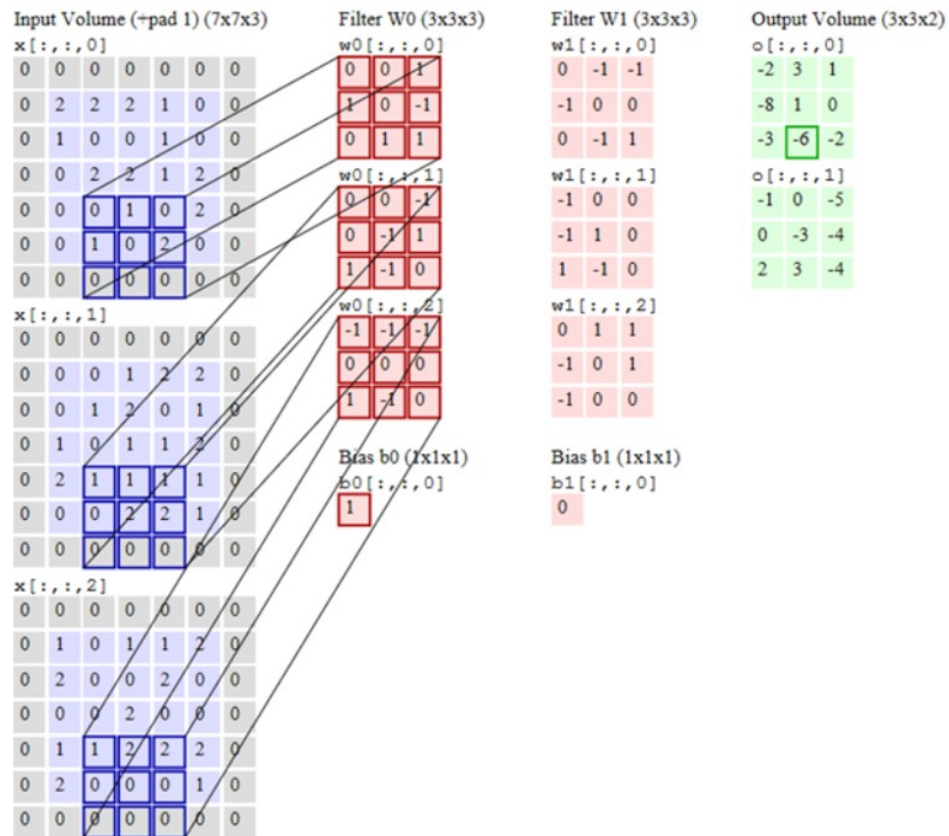
Membentuk ulang fitur (reshape fitur map) menjadi sebuah vektor agar bisa kita gunakan sebagai input dari fully-connected layer

b) Fully-connected

Pada fully connected lapisan terhubung sepenuhnya akan menghitung skor kelas. Setiap neuron dalam lapisan ini akan terhubung ke semua angka dan volume

c) Softmax

Fungsi Softmax menghitung probabilitas dari setiap kelas target atas semua kelas target yang memungkinkan dan akan membantu untuk menentukan kelas target untuk input yang diberikan. Keuntungan utama menggunakan Softmax adalah rentang probabilitas output dengan nilai 0 hingga 1, dan jumlah semua probabilitas akan sama dengan satu. Jika fungsi softmax digunakan untuk model multi-klasifikasi, dia akan mengembalikan peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas tinggi. Softmax menggunakan eksponensial ( $e$ -power) dari nilai input yang diberikan dan jumlah nilai eksponensial dari semua nilai dalam input. Maka rasio eksponensial dari nilai input dan jumlah nilai eksponensial adalah output dari fungsi softmax.

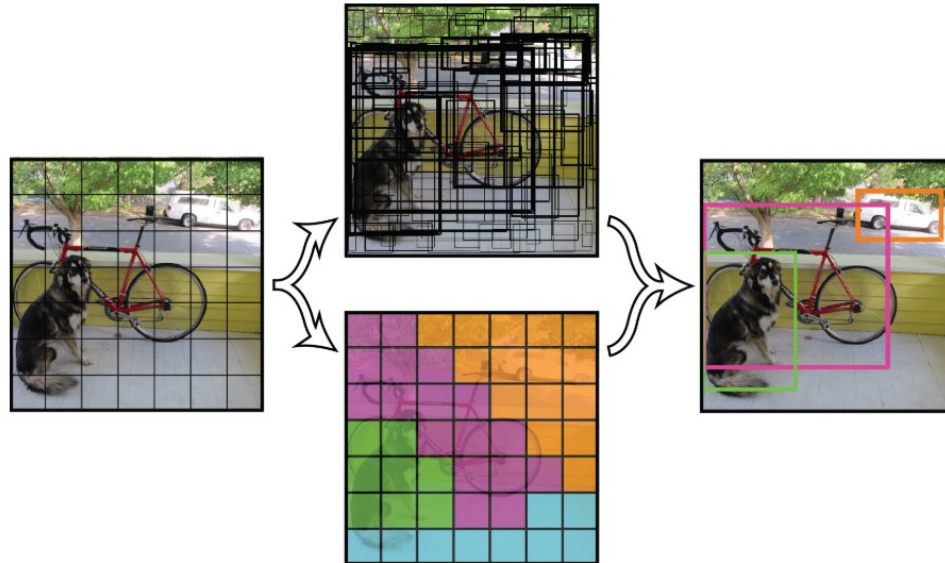


Gambar 12. Langkah menghitung CNN

## 6) YOLO

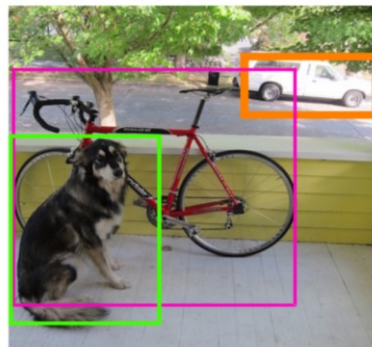
Algoritma YOLO merupakan *Real Object Detection* yang baru-baru ini sangat populer untuk dikembangkan. Kebanyakan sistem deteksi sebelumnya menggunakan pengklasifikasian atau *localizer* untuk melakukan deteksi dengan menerapkan model ke gambar di beberapa lokasi dan skala dan memberi nilai pada gambar sebagai bahan untuk pendeteksian. YOLO menggunakan pendekatan yang sangat berbeda dengan algoritma sebelumnya, yakni menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi

wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasikan sebagai objek atau bukan.



Gambar 13. Gambar dengan wilayah pembatas

Pada bagian akhir akan dipilih kotak pembatas dengan nilai confidence yang paling tinggi untuk di jadikan sebagai pemisah objek satu dengan yang lain seperti pada gambar berikut



Gambar 14. Hasil deteksi menggunakan YOLO

YOLO lebih unggul di bandingkan sistem berbasis classifier terlihat pada seluruh gambar pada waktu pengujian sehingga prediksinya di informasikan oleh konteks global dalam gambar. Sehingga membuat prediksi dengan evaluasi jaringan tunggal tidak seperti sistem R-CNN yang memerlukan ribuan gambar. Sehingga kinerja YOLO menjadi sangat cepat, 1000x lebih cepat dari R-CNN dan 100x lebih cepat dari Fast R-CNN (Redmon and Farhadi, n.d.).

Prinsip deteksi pada YOLO input gambar di bagi kedalam  $S \times S$  grid secara umum. Pada tiap grid cell misalkan  $nB=2$  terdapat  $B$  yang berisi 5 nilai yaitu lokasi koordinat  $x, y$  ukuran lebar dan tinggi bounding box dalam  $w, y$  dan nilai confidence  $(x, y, w, h, c)$  terhadap  $nB$  menyatakan banyaknya bounding box. Tiap grid memprediksi kemungkinan dari nilai confidence ( $C$ ). Kemungkinan reflek skor confidence dari sebuah model termasuk target objek dan akurasi dari prediksi box deteksi.

$$C(\text{Object}) = \text{Pr}(\text{Object}) \times \text{IOU}(\text{Pr ed}, \text{Truth}) \quad (1)$$

IOU merupakan angka overlap dari kandidat bound secara umum dan ground truth bound, rasio dari interseksi dan unionnya

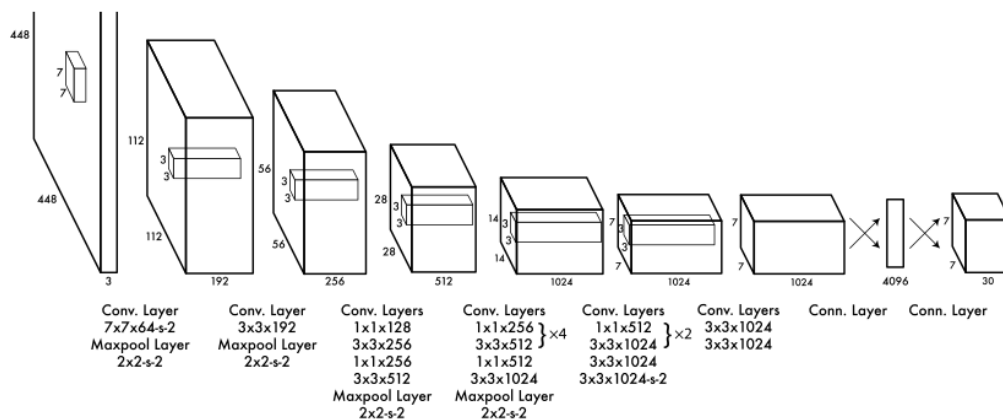
$$\text{IOU}(\text{Pr ed}, \text{Truth}) = \frac{\text{area}(\text{box}_{\text{truth}}) \cap \text{area}(\text{box}_{\text{prod}})}{\text{area}(\text{box}_{\text{truth}}) \cup \text{area}(\text{box}_{\text{prod}})} \quad (2)$$

Setelah mendapat confidence dari tiap prediksi box, nilai terendah prediksi box dihapus dengan nilai treshold dan kemudian penekanan non



maksimum di bentuk dari bounding box yang tersisa (Yang and Jiachun, 2018).

Pada YOLO menerapkan model sebagai *convolutional neural network*. Pada awalnya layer *convolutional* pada fitur ekstraksi jaringan dari gambar sementara lapisan sepenuhnya terhubung memprediksi probabilitas dan koordinat. Arsitektur YOLO terinspirasi dari GoogLeNet untuk model klasifikasinya. Yolo memiliki 24 lapisan *convolutional* di ikuti oleh 2 lapisan yang terhubung sepenuhnya. Tetapi dibandingkan GoogLeNet modul permulaan yang digunakan YOLO lebih simpel menggunakan reduksi layer 1x1 diikuti 3x3 layer convolutional. Desain network Yolo sebagaimana pada gambar berikut



Gambar 15. Arsitektur Algoritma YOLO

## B. Penelitian Terkait

Beberapa penelitian terkait sistem deteksi larva dengan beberapa metode seperti pada beberapa penelitian:

1. Johan Musaeus Bruun et.al meneliti tentang mendeteksi Telur dari cacing cambuk babi kecil *Trichuris suis*. Metode yang digunakan berbasis visi untuk mendeteksi dan mengklasifikasi telur parasit *T.Suis*. Deteksi adalah berdasarkan filter yang cocok dan klasifikasi dilakukan dengan menggunakan analisis diskriminan linear dan kuadratik pada seperangkat biologis fitur yang diilhami, termasuk berbasis autokorelasi anisotropi longitudinal dan intensitas hamburan rata-rata di bawah iluminasi lapangan gelap. metode yang diusulkan mencapai crossvalidated tingkat klasifikasi sekitar 93%(Bruun et al., 2012).
2. Dedy Harto dkk, meneliti tentang penyakit *whitespot* (Bintik putih) pada udang. Dalam penelitian ini di lakukan pengembangan perangkat lunak untuk mendeteksi *white spot* pada udang dengan menggunakan pengolahan citra digital dan Jaringan Syaraf Tiruan(JST). Proses pengolahan citra digital dimulai dari proses cropping, grayscale, histogram yang menghasilkan nilai-nilai piksel grayscale yang di gunakan sebagai masukan pada JST(Harto and Sarosa, 2012).
3. Guo Jiau et.al Dalam meneliti tentang metode untuk menemukan kepala dan ekor ikan zebra di latar belakang statis oleh menggabungkan reduksi latar belakang dan metode diferensial bingkai. Hasil eksperimen menunjukkan bahwa metode ini dapat mendeteksi larva akurat ketika

menghasilkan kejang. Kemudian lintasan dua dimensi larva diperoleh. Metode ini tinggi akurasi, kecepatan tinggi, dan mudah diimplementasikan(Guo et al., 2014).

4. Xianmu Zheng et.al meneliti tentang deteksi dini dan prediksi hama dan merancang program yang dapat mendeteksi larva geometrid secara otomatis menggunakan teknologi pemrosesan gambar. Pertama sesuai dengan karakteristik warna larva, mengekstrak kandidat larva coklat dari gambar dengan latar daun hijau. Menggunakan operasi morfologis untuk memperbaiki daerah kandidat. Kemudian menggunakan elliptic fourier transform untuk mewakili kontur kandidat larva. Representasinya adalah sebuah vektor. Setelah itu kami membandingkan semua vektor kontur yang mungkin dari larva dengan vektor kontur larva yang dipilih. Menurut jarak antara vektor, kami menentukan apakah kontur adalah kontur larva yang benar. Hasil percobaan menunjukkan bahwa kami. Metode efektif dapat mengekstrak larva geometri di kebun the gambar. Yang dirancang Program mampu mendeteksi 81,3% larva melengkung. Larva tingkat deteksi dapat diterima dalam aplikasi pertanian teknik(Zheng and Cai, 2016).
5. Andres at.al meneliti tentang sistem pemantauan otomatis untuk pendeteksian larva Kerang Peru dalam gambar mikroskopis disajikan. Algoritma pemrosesan gambar dan prosedur statistik miliki telah diterapkan untuk mendeteksi dan menghitung larva yang terkandung dalam gambar mikroskopis serta untuk mengukur area larva individu.

Algoritma sederhana telah diusulkan untuk menilai larva karakteristik. Prosedur ini termasuk pemrosesan gambar dan analisis statistik. Sistem dapat menghitung larva dari gambar mikroskopis dengan akurasi 95% dibandingkan dengan penghitungan manual (Flores et al., 2008).

6. Rosa Andri meneliti tentang rancang bangun penghitung benih ikan menggunakan *binary thresholding* pada raspberry pi secara *real time*. Selama ini petani ikan masih melakukan perhitungan benih secara manual yaitu dengan menghitung satu per satu atau menggunakan volume (gelas). Sehingga selain memakan waktu yang lama, benih ikan terkadang stress dikarenakan perhitungan yang masih manual. Penelitian ini mendesain dan mengembangkan alat yang mampu menghitung benih ikan dengan mengimplementasikan pengolahan citra sebagai solusi untuk mengatasi permasalahan para petani ikan. Sistem yang dirancang dan diimplementasikan menggunakan *HTML*, *Python*, serta pengolahan citra yang menggunakan metode *Thresholding*, *Morphology*, serta pelabelan. Sistem ini diterapkan secara *real time*, serta dapat menghitung objek yang mendekati perhitungan yang sebenarnya. Sistem ini telah diuji menggunakan 4 data set yaitu benih yang diuji tiap kelipatan 10 dan berakhir pada pengujian 40 benih ikan. Tingkat keakuratan tertinggi mencapai 99.9977 % untuk pengujian perhitungan 40 benih (A et al., 2017)
7. Sheryl et.al melakukan penelitian untuk menghitung ikan fingerlings menggunakan teknik image processing dan menyelidiki efektifitas *Convolutional Neural Network (CNN)* pada deteksi dan akurasi

perhitungannya. Teknik yang di usulkan dengan mengetest pada empat ukuran yang berbeda. Nilai ambang batas ditetapkan untuk meningkatkan tingkat efisiensi deteksi ikan dan akurasi penghitungan. Metode eksperimental dan menangkap gambar dari jari itu dilakukan di Biro Perikanan dan Sumber Daya Perairan (BFAR) Laboratorium Pembenihan Ikan Nila dan Kesehatan Ikan di Kalimantan Barcenaga, Naujan, Oriental Mindoro, Filipina. 2400 gambar bibit nila digunakan dalam pelatihan dan 1.600 gambar ditahap pengujian(Lainez and Gonzales, 2019b).

### **C.State Of The Art**

Pada tabel berikut menampilkan perkembangan penelitian image processing untuk mendeteksi larva maupun mendeteksi objek dengan ukuran kecil atau objek yang memiliki kemiripan proses image processing seperti larva kepiting atau menyerupai objek larva kepiting. Pada paper Alejandra Sanchez Ortiz et/al telah di bahas tentang mendeteksi siklus larve nyamuk dengan menggunakan jaringan syaraf tiruan adapun akurasi yang di capai hingga 97% dan Dedi Harto dkk pada penelitian mendeteksi white spot menggunakan jaringan syaraf tiruan juga mencapai akurasi hingga 93%. Berdasarkan kemiripan objek baik di larva maupun pada objek yang berukuran kecil seperti white spot pada udang tersebut maka diputuskan untuk menggunakan metode Jaringan Syaraf Tiruan.

Tabel 1. State of the Art

No	Judul	Penulis	Penerbit /Tahun	Tujuan	Metode	Hasil
1	Detection and Classification of parasite eggs for use in helminthic Therapy	Johan Musaeus Bruun, Christian M.O.Kapel, Jens Michael Cartensen	IEEE/ 2012	mendeteksi Telur dari cacing cambuk babi kecil Trichuris suis	Detection, Featur Extraction: Longitudinal anisotropy dan Mean Scattering Intensity, Clasification	Detecti on 98.9% Clasific ation 92.7%
2	Aplikasi Jaringan Syaraf Tiruan untuk Mendeteksi <i>White Spot</i>	Dedy Harto, M. Sarosa, Wijono dan Suprpto	<i>Jurnal EECCIS Vol. 6, No. 1, Juni 2012</i>	meneliti tentang penyakit <i>whitespot</i> (Bintik putih) pada udang	JST Backpropagasi	93%
3	Trajectory Trackingn of Spasm-Oriented Zebrafish larvae	Guo Jiayu, Zhao Shasha, Shu Mao, Yan Zhongguang, Sun Mingzhu, Zhao Xin, Feng Xizeng	National Natural Science Foundati on of China(N	metode untuk menemukan kepala dan ekor ikan zebra di latar belakang statis oleh menggabungkan	reduksi latar belakang dan metode diferensial bingkai	99%

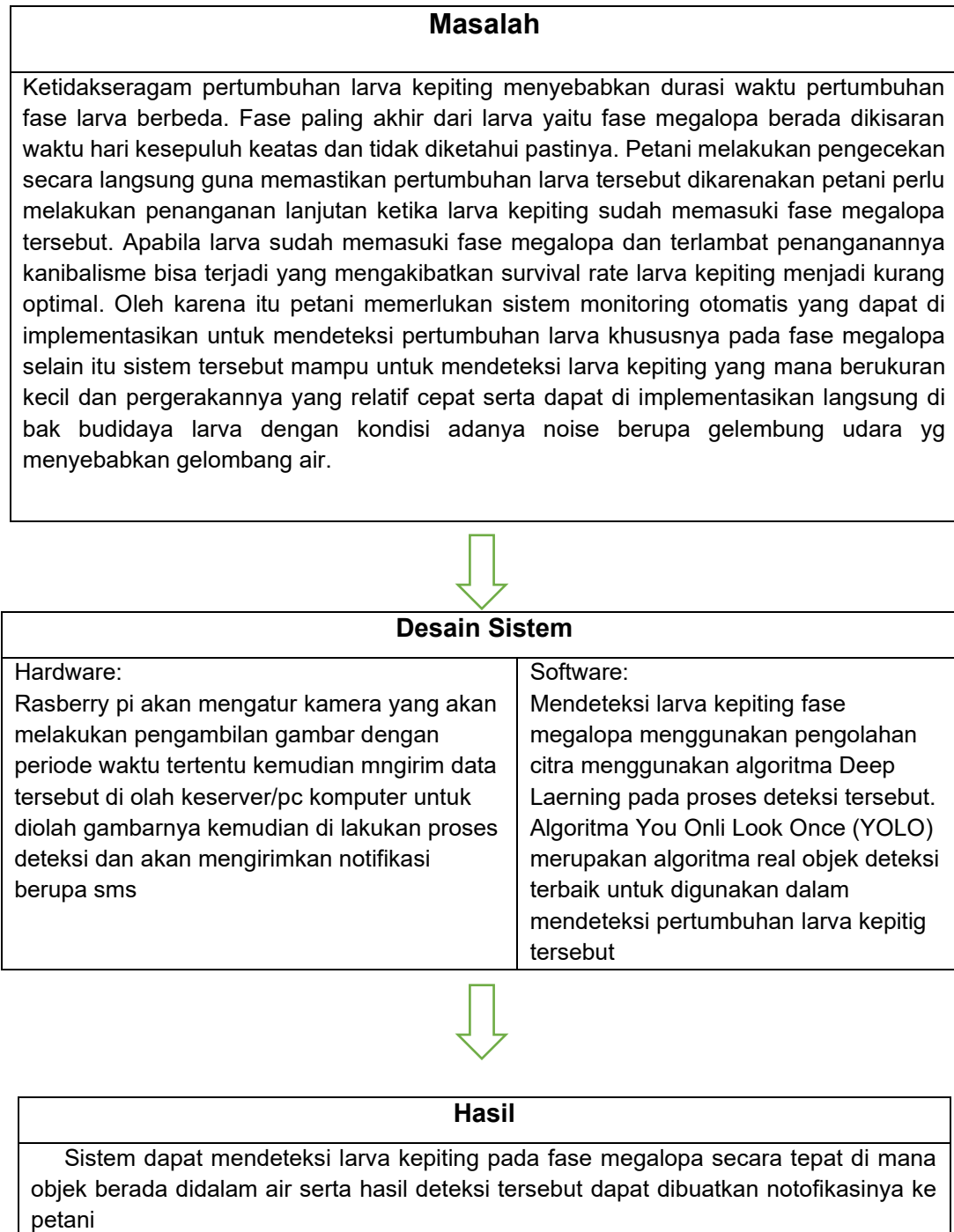
No	Judul	Penulis	Penerbit /Tahun	Tujuan	Metode	Hasil
			SFC) 2014	reduksi latar belakang dan metode diferensial bingkai		
4	Geometrid Larvae Detection Using Contour Feature	Xianmu Zheng, Rongtai Cai	IEEE/ 2016	deteksi dini dan prediksi hama dan merancang program yang dapat mendeteksi larva geometrid secara otomatis menggunakan teknologi pemrosesan gambar	operasi morfologis untuk memperbaiki daerah kandidat. elliptic fourier transform untuk mewakili kontur kandidat larva. Representasinya: vektor. membandingkan semua vektor kontur yang mungkin dari larva dengan vektor kontur larva	81.3%

No	Judul	Penulis	Penerbit /Tahun	Tujuan	Metode	Hasil
					yang dipilih. Menurut jarak antara vektor,	
5	Peruvian Scallop Larvae Counting System Using Image Processing Techniques	Andrés Flores, Pedro Crisóstomo, Jean López Sección de Electricidad y Electrónica Pontificia Universidad Católica del Perú Lima, Peru	2008	Mendeteksi larva peruvian scallop	Image processing	95%
6	Rancang Bangun Penghitung Benih Ikan Menggunakan <i>Binary Thresholding</i>	Rosa Andrie A. Irawati Nurmala Sari	Jurnal Informati ka Polinem a /2017	mendesain dan mengembangka n alat yang mampu menghitung benih ikan	<i>Thresholding</i> , <i>Morphology</i> , serta pelabelan	99%



No	Judul	Penulis	Penerbit /Tahun	Tujuan	Metode	Hasil
	Pada Raspberry Pi Secara <i>Real Time</i>	, Vivid Ichtarosa Arinda		dengan mengimplementasikan pengolahan citra sebagai solusi untuk mengatasi permasalahan para petani ikan		
7	Automated Fingerlings Counting Using Convolutional Neural Network	Sheryl May D. Lainez, Dennis B. Gonzales	IEEE/ 2019	menghitung ikan fingerlings menggunakan teknik image processing dan menyelidiki efektifitas <i>Convolutional Neural Netwok (CNN)</i> pada deteksi dan akurasi perhitungannya	CNN	99%

## D.Kerangka Pikir



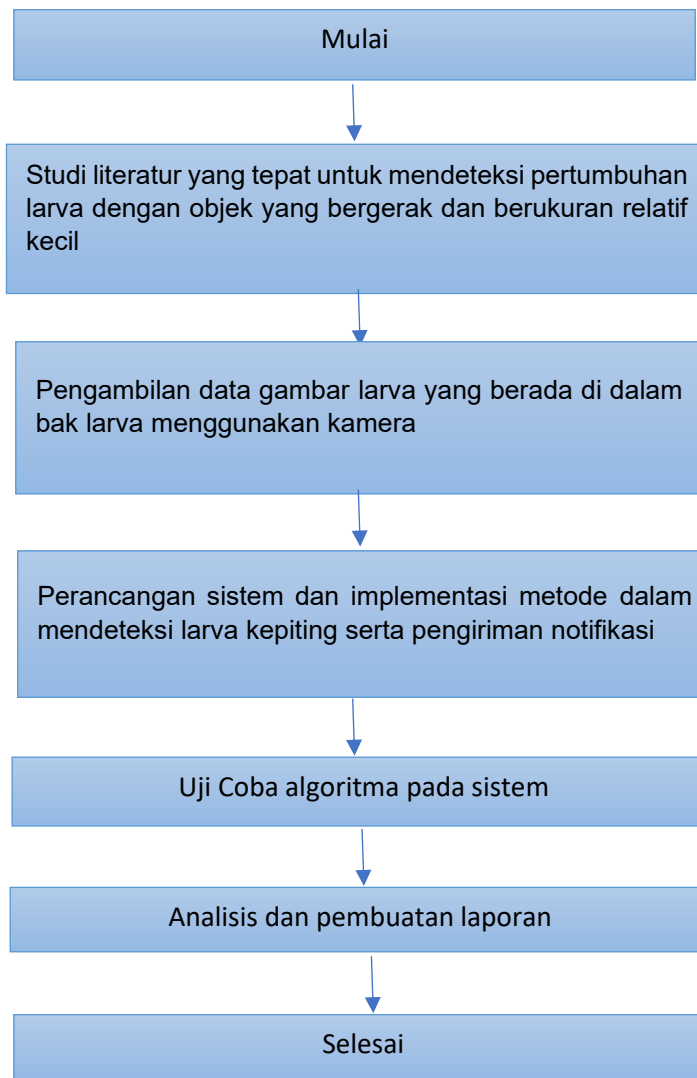
Gambar 16. Kerangka pikir penelitian

## BAB III

### METODE PENELITIAN

#### A.Tahap Penelitian

Tahapan penelitian akan di uraikan pada gambar berikut ini



Gambar 17. Tahapan penelitian

Dari gambar 17 secara garis besar di uraikan sebagai berikut:

1. Studi literatur penelitian terkait metode deteksi yang digunakan pada deteksi larva kepiting di uraikan berdasarkan referensi yang bersumber dari jurnal atau paper penelitian, buku, artikel dan laporan penelitian
2. Pengambilan data gambar larva yang berada di dalam bak pemeliharaan menggunakan kamera webcam
3. Perancangan sistem dilakukan dengan membuat diagram sistem mulai dari proses deteksi mulai dari penyiapan data gambar yang akan digunakan sebagai data latih serta pengiriman notifikasi
4. Uji coba algoritma pada sistem dilakukan dengan menguji algoritma yang digunakan untuk mengetahui akurasi dan efektifitas algoritma yang digunakan
5. Setelah dilakukan proses uji coba algoritma atau metode yang digunakan kemudian dilakukan analisis. Tahapan ini menganalisis setiap paramater yang digunakan dengan menghitung tingkat akurasinya. Proses akhir yang dilakukan yaitu pembuatan laporan publikasi dan laporan akhir magister.

### **B.Waktu Dan Lokasi Penelitian**

#### 1. Waktu

Waktu Penelitian akan dilaksanakan selama 10 bulan dimulai pada bulan Januari hingga November 2019.

## 2. Lokasi

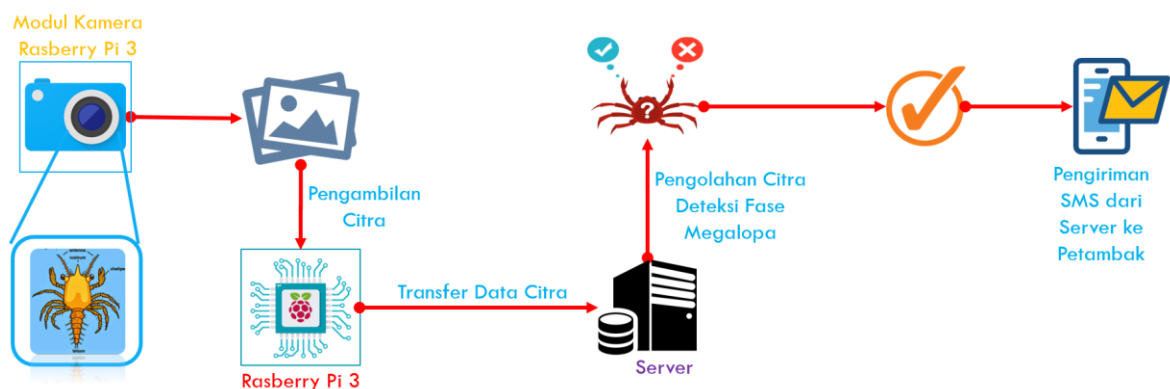
Penelitian dilakukan di Fakultas teknik Elektro Universitas Hassanuddin dan Balai Pembenihan Budidaya Air Payau (BPBAP) Takalar Kab. Takalar Propinsi Sulawesi Selatan

### C.Jenis Penelitian

Jenis penelitian ini merupakan penelitian eksperimental yang bersifat analisis sehingga dari ruang lingkup masalah dapat dilakukan dengan metode studi pustaka (*library research*), metode pengumpulan data (*field research*) dan perancangan sistem serta analisis.

### D.Perancangan Sistem

Rancangan sistem pada penelitian ini mengikuti beberapa tahapan yang digambarkan sebagai berikut:



Gambar 18. Rancangan Sistem Deteksi Larva Fase Megalopa

Pada gambar 18, terdapat 4 tahapan proses yang dilakukan yaitu tahapan pengambilan gambar, tahapan transfer data, tahapan pengolahan citra untuk deteksi larva fase meaglopa, dan tahapan pengiriman notifikasi. Sebelum semua tahapan pada perancangan sistem diatas, terdapat satu tahapan penting yang dilaksanakan yaitu tahapan proses pelatihan data gambar larva yang mana hasil dari proses pelatihan ini akan dipakai pada tahapan pengolahan citra yang dilakukan oleh komputer server.

Penjelasan setiap tahapan pada rancangan sistem termasuk tahapan pelatihan dijelaskan sebagai berikut:

### **1. Pengambilan Gambar**

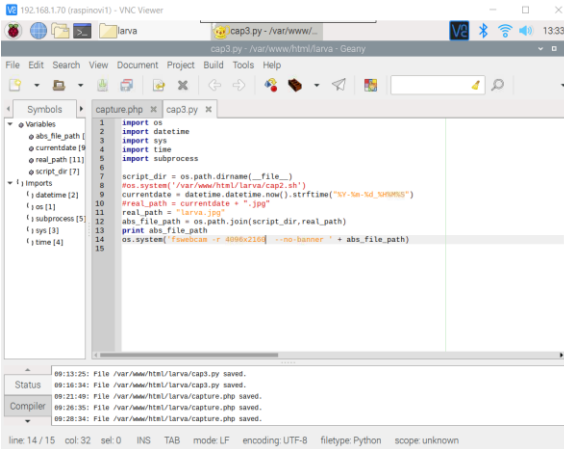
Pada penelitian ini proses pengambilan gambar telah diujicoba menggunakan beberapa kamera digital diantaranya sony RX100, kamera gopro hero 5, kamera DSLR Canon EOS 80D dan kamera Logitech Brio. Pada saat digunakan pada uji coba pengambilan data tiap kamera tersebut memiliki keunggulan dalam proses pengambilan gambar, tetapi dari beberapa kamera tersebut seperti Sony RX 100, kamera gopro hero 5, kamera DSLR Canon EOS 80D memiliki permasalahan tersendiri dalam pengambilan data, yang menjadi masalah adalah ketika gambar yang diambil hendak ditransfer secara otomatis ke komputer server per satuan waktu. Kamera-kamera tersebut hanya menyediakan satu jalur komunikasi melalui kabel yang mana jalur untuk transfer data menjadi satu dengan jalur untuk catu daya.

Pada pengambilan data set mulai dari larva zoea 1 hingga megalopa memerlukan waktu kurang lebih 15 hari, dengan kondisi dimana jalur untuk transfer data dan catu daya yang hanya memiliki satu jalur, tidak memungkinkan digunakan pada proses pengambilan gambar di lapangan karena kamera-kamera tersebut mempunyai kapasitas daya yang terbatas jika tidak dilakukan pengisian baterai. Sehingga pengambilan data gambar menggunakan kamera webcam Logitech BRIO 4K Ultra HD. Penggunaan kamera tersebut dipilih dengan pertimbangan:

1. Kamera yang digunakan merupakan kamera yang dapat berkomunikasi dan dapat di kontrol oleh Raspberry Pi 3
2. Kamera yang digunakan memiliki jalur sumber daya listrik yang jadi satu dan lebih sederhana sehingga tidak mengganggu proses pengambilan gambar
3. Kamera yang digunakan memenuhi kriteria mempunyai fokus mode macro dikarenakan objek larva berukuran relatif kecil
4. Kamera yang digunakan mempunyai kecepatan 60 fps, pertimbangan kecepatan fps di karenakan objek larva tersebut merupakan objek dengan pergerakan yang cukup cepat.

Pada tahapan ini, kamera webcam terkoneksi dengan Raspberry Pi untuk proses pengambilan gambar. Pada Raspberry Pi, proses capture gambar menggunakan library fswebcam yang diatur untuk melakukan perekaman gambar dengan resolusi 4096x2160 piksel dan menghasilkan

gambar dengan ekstensi \*.jpg. hasil perekaman gambar disimpan pada memory Raspberry Pi dengan nama larva.jpg. Ssetiap pengambilan gambar baru, hasil gambar akan menimpa file gambar lama dengan penamaan file yang sama. Hal ini untuk mencegah terjadi penuhnya kapasitas memory Raspberry Pi.



```

1 import os
2 import datetime
3 import sys
4 import time
5 import subprocess
6
7 script_dir = os.path.dirname(__file__)
8 os.system("/var/www/html/larva/cap2.sh")
9
10 #currentdate = datetime.datetime.now().strftime("%Y-%m-%d_%H%M%S")
11 #real_path = currentdate + ".jpg"
12 real_path = "larva.jpg"
13 abs_file_path = os.path.join(script_dir, real_path)
14 print abs_file_path
15 os.system["fusercan -r 4886x216[ ...no-banner ...] = abs_file_path"]

```

09:13:25: File /var/www/html/larva/cap3.py saved.  
09:16:34: File /var/www/html/larva/cap3.py saved.  
09:21:49: File /var/www/html/larva/capture.php saved.  
09:26:30: File /var/www/html/larva/capture.php saved.  
09:28:34: File /var/www/html/larva/capture.php saved.

line 14 / 15 col: 52 sel: 0 INS TAB mode LF encoding UTF-8 filetype Python scope: unknown

Gambar 19. Code Pengambilan Gambar di Raspberry Pi

## 2. Transfer Data

Proses pengolahan dan deteksi gambar pada penelitian ini dilakukan pada komputer server yang terhubung menggunakan jaringan LAN dengan Raspberry Pi. Komputer server akan melakukan komunikasi ke Raspberry Pi berupa request pengambilan file gambar, kemudian raspberry pi melakukan proses perekaman, selanjutnya komputer server akan melakukan proses download gambar hasil perekaman sesuai alamat yang diberikan pada respon dari raspberry pi. Proses transfer data ini



menggunakan syntaks php yang dimaksudkan untuk dapat diakses melalui port http.

```

1 <?php
2 //Soutput = shell_exec('/usr/bin/fswebcam -S 1 -r 640x480 -q --no-banner /var/www/html/larva/foto.);
3 $output = shell_exec('python /var/www/html/larva/cap3.py');
4 $msg = array("path"=>$output, "fn"=>basename($output), "url"=>$SERVER['REQUEST_URI']);
5 //echo str_replace('/var/www/html/larva/', '', $output);
6 echo json_encode($msg);
7
8

```

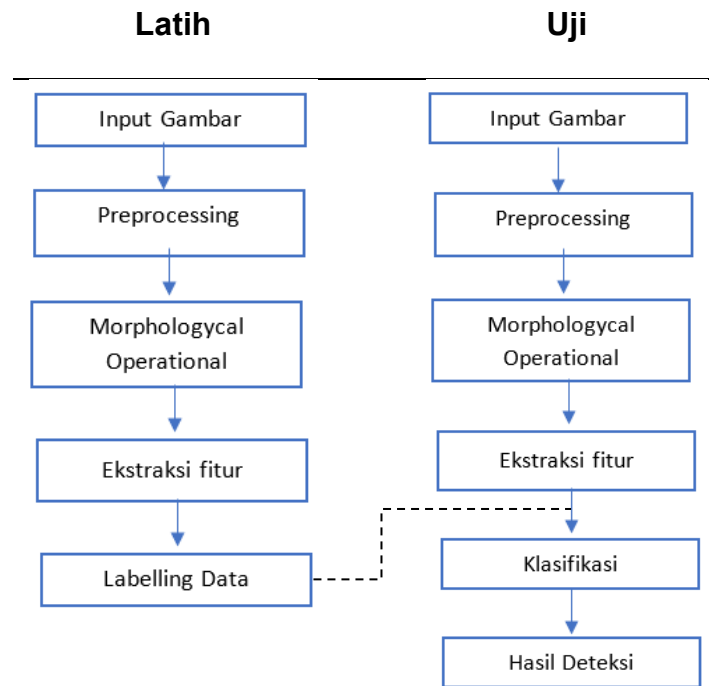
Gambar 20. Code Transfer data

### 3. Sistem Deteksi Larva

Pada awal pengembangan sistem deteksi, teknik pengolahan citra menggunakan algoritma SVM untuk proses deteksi larva, yang kemudian diganti dengan menggunakan algoritma YOLO v3.

#### a. Deteksi menggunakan Algoritma SVM

Tahapan penggunaan pengolahan citra menggunakan algoritma svm yang dilakukan pada penelitian ini digambarkan sebagai berikut:



Gambar 20. Tahapan Deteksi Larva menggunakan algoritma svm

Penjelasan dari tiap tahapan pada gambar diatas adalah sebagai berikut:

1) Input Gambar

Gambar yang digunakan sebagai gambar input adalah gambar larva dengan resolusi 4096x2160 piksel yang kemudian dirize menjadi gambar dengan ukuran 600x600 piksel.

2) Pre processing

Pada tahapan ini, dilakukan 2 proses pengolahan gambar yaitu proses *grayscale* (citra abu-abu) dan proses *sharpening* (penajaman gambar).

a) *Grayscale*

Proses grayscale bertujuan untuk mengubah intensitas warna RGB(tiga nilai) menjadi 1 skala tunggal yaitu warna keabu-abuan. Untuk konversi grayscale, setiap pixelRGB yang memiliki 3 bobot nilai akan di kalikan masing-masing dengan komponen luminance(Y) yang memiliki bobot terhadap beberapa komponen intensitas warna yaitu red= 0.299, green =0.587 dan blue 0.114 (Saravah,2010).

$$(\text{Red} \cdot 0.299) + (\text{Green} \cdot 0.587) + (\text{Blue} \cdot 0.14)$$

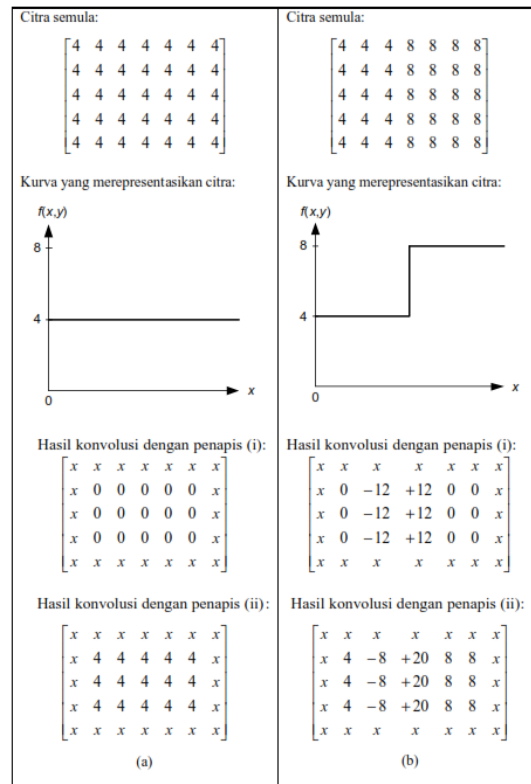
Hasil dari persamaan rumus tersebut mendapat satu nilai yang merupakan nilai terhadap pixel yang baru dengan keabu-abuan. Process selanjutnya yaitu gaussian blur yang bertujuan untuk menghilangkan noise pada citra. Hasil pre processing pada proses grayscale dan Gaussian blur dapat dilihat pada gambar berikut:



Gambar 21. Hasil Pengolahan Citra Grayscale dan Gaussian Blur

b) *Sharpening*

Sharphening atau penajaman citra bertujuan memperjelas tepi pada objek di dalam citra. Penajaman citra merupakan kebalikan dari operasi pelembutan citra karena operasi ini menghilangkan bagian citra yang lembut. Operasi penajaman dilakukan dengan melewatkan citra pada high-pass filter) yang akan memperkuat komponen berfrekuensi tinggi(misalkan tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat lebih tajam di bandingkan sekitarnya. Karena penajaman citra lebih berpengaruh pada tepi (edge) objek, maka penajaman citra sering disebut juga penajaman tepi (*edge sharpening*) atau peningkatan kualitas tepi (*edge enhancement*). Karena koefisien penapis mengandung nilai negatif, maka konvolusi mungkin saja menghasilkan pixel bernilai negatif. Meskipun intensitas bernilai negatif menarik, tetapi kita tidak dapat menampilkannya. Untuk alasan terakhir ini, implementasi konvolusi men-set nilai negatif menjadi nilai 0. Cara lainnya adalah dengan mengambil nilai mutlaknya atau menskalakan semua nilai-nilai pixel secara menaik sehingga nilai yang paling negatif menjadi 0.



Gambar 22. Implementasi Konvolusi



Gambar 23. Hasil Pengolahan implementasi konvolusi

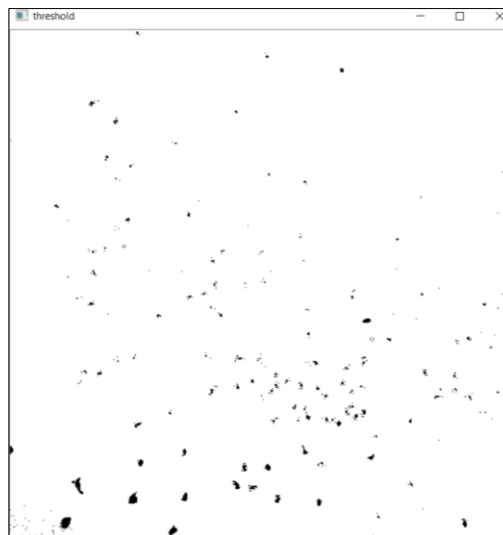
### 3) Morphological operational

Pada tahapan ini dilanjutkan dengan operasi thresholding:

Pada operasi thresholding, nilai intensitas pixel dipetakan ke salah satu dari dua nilai,  $a_1$  atau  $a_2$  berdasarkan nilai ambang (*thresholding*)  $T$ :

$$f(x, y)' = \begin{cases} a_1, & f(x, y) < T \\ a_2, & f(x, y) \geq T \end{cases}$$

Jika  $a_1 = 0$  dan  $a_2 = 1$ , maka operasi pengambangan mentransformasikan citra hitam-putih ke citra biner. Dengan kata lain, nilai intensitas pixel semula dipetakan ke dua nilai saja: hitam dan putih. Nilai ambang yang dipakai dapat berlaku untuk keseluruhan pixel atau untuk wilayah tertentu saja (berdasarkan penyebaran nilai intensitas pada wilayah tersebut).



Gambar 24. Proses tresholding

#### 4) Ekstraksi Fitur

Tahapan selanjutnya yaitu menentukan daerah kandidat atau ROI (Region of Interest). Daerah kandidat di tandai dengan boundingbox yang kemudian dihitung posisi centroid dari bounding box tersebut. Fitur yang di gunakan sebagai pembeda antara class fase megalopa dan non megalopa berupa ukuran dari tiap bounding box yang kemudian di olah untuk mencari nilai eigen value. Nilai eigen value tersebut nantinya yang akan digunakan sebagai nilai dalam proses klasifikasi.

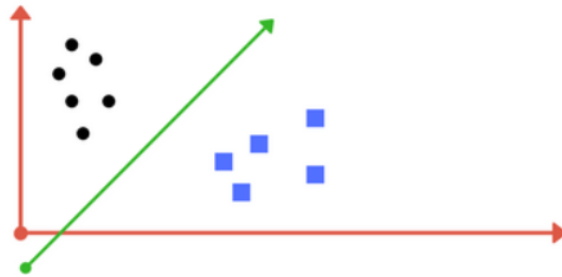
Berikut gambar dari wilayah kandidat atau ROI beserta nilai eigen value dan value vektor dari image larva yang telah dilakukan proses pengolahan citra dari tahap-tahap sebelumnya.



Gambar 25. Ekstraksi fitur

### 5) Klasifikasi menggunakan Algoritma SVM

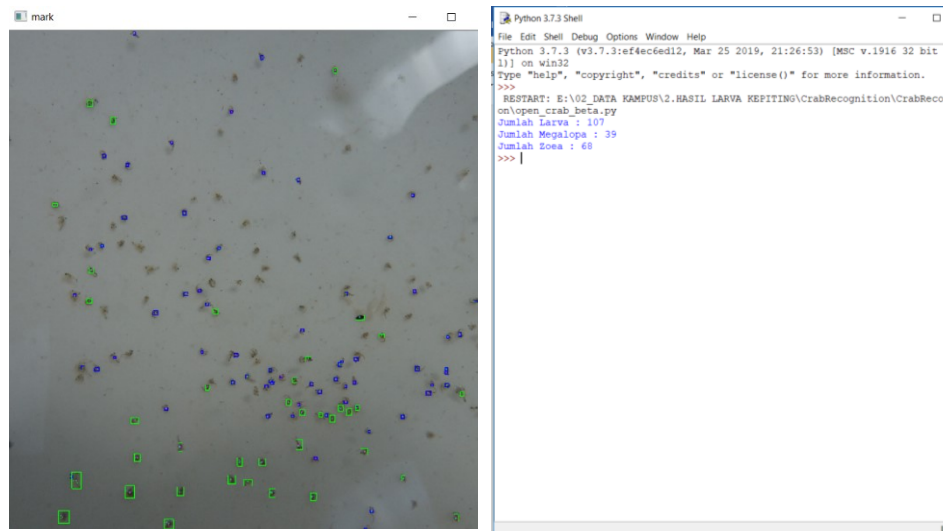
Super Vector Machine (SVM) adalah Classifier diskriminatif yang secara formal didefinisikan oleh hyperplane pemisah. Dengan kata lain, diberikan data pelatihan berlabel (supervise learning), algoritma menghasilkan hyperplane optimal yang mengkategorikan contoh baru. Dalam dua ruang dimensi hyperplane ini merupakan sebuah garis yang membagi sebuah plane menjadi dua bagian dimana disetiap kelas terletak di kedua sisi. Algoritma svm merupakan bagian algoritma mechine learning.



Gambar 26. Grafik svm

Pada penelitian ini digunakan algoritma svm, dengan nilai eigenvalue dan value vektor dari tahap sebelumnya kemudian dimasukkan kedalam matriks dari algoritma svm sehingga diperoleh hasil seperti gambar berikut ini:





Gambar 27. Hasil klasifikasi menggunakan svm

#### 6) Hasil Deteksi

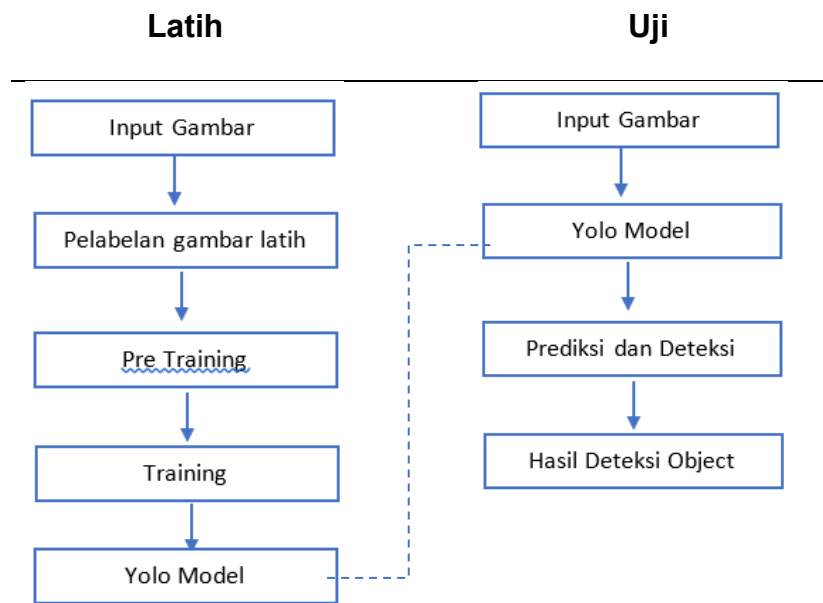
Hasil yang di peroleh menggunakan rancangan awal dengan mengujikan dibeberapa gambar tingkat akurasi yang diharapkan masih belum maksimal. Pada saat pengujian sistem tersebut masih banyak terdapat kesalahan deteksi larva dimana larva fase non megalopa yang terdeteksi sebagai megalopa. Dalam menentukan nilai yang diklasifikasikan menggunakan algoritma (Super Vektor Machine) SVM menggunakan nilai eigenvalue dan valuevektor berdasarkan daerah kandidiat objek sehingga didapatkan rentang nilai yang dijadikan penentu apakah fase tersebut fase megalopa atau bukan megalopa. Pada gambar uji yang dilakukan terjadinya kesalahan dalam mendeteksi tersebut di karenakan nilai ROI (Nilai eigen value dan value vektor) dari

gambar uji tersebut tidak tepat sama sehingga mengakibatkan kesalahan deteksi antara megalop dan bukan megalopa. Sehingga pada penelitian dilakukan perancangan ulang sistem menggunakan metode deteksi yang lain.

b. Algoritma YOLOv3

Perubahan penggunaan algoritma pada penelitian ini yang dari yang awalnya menggunakan algoritma Super Vector Mechine (SVM) sampai pada penggunaan algoritma YOLOv3 dimaksudkan untuk mencapai hasil akurasi deteksi larva fase megalopa yang maksimal. Algoritma YOLO sendiri telah berkembang mulai sejak versi pertama, kemudian dimaksimalkan menjadi versi kedua, sampai akhirnya saat ini muncul YOLO versi ketiga yang dibuat lebih cepat performa kinerjanya dalam melakukan proses pelatihan data gambar dan prediksi multi objek.

Tahapan proses penggunaan algoritma YOLOv3 pada penelitian ini dapat dilihat pada gambar 28.



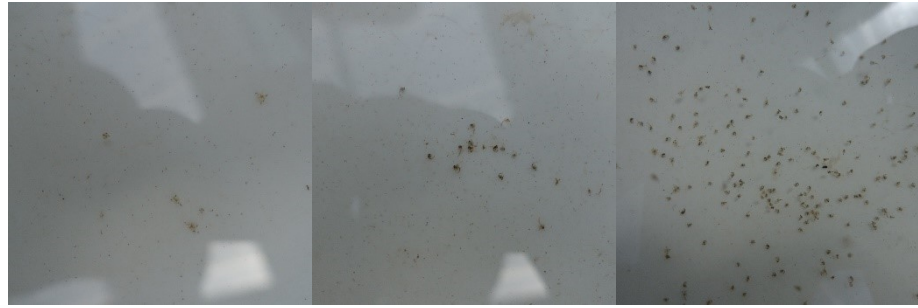
Gambar 28. Tahapan Penggunaan Algoritma YOLOv3

Penjelasan untuk setiap tahapan pada gambar diatas adalah sebagai berikut:

#### 1) Penyiapan Gambar

Gambar yang akan digunakan sebagai data latih sebanyak 200 gambar. Gambar yang akan di ujikan terdiri dari gambar larva kepiting fase megalopa dan bukan fase megalopa. Gambar yang digunakan memiliki ukuran 4096 x 2160 piksel. Pada proses training ukuran gambar akan diubah menjadi 416 x 416 piksel.

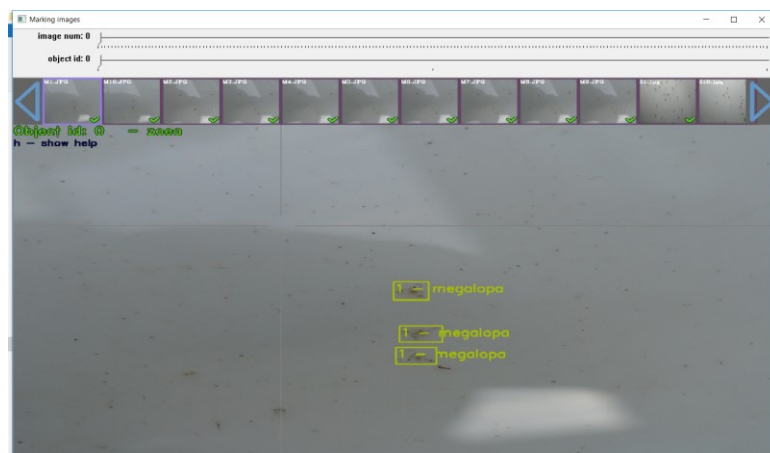
Berikut ini contoh beberapa gambar yang dilatih:



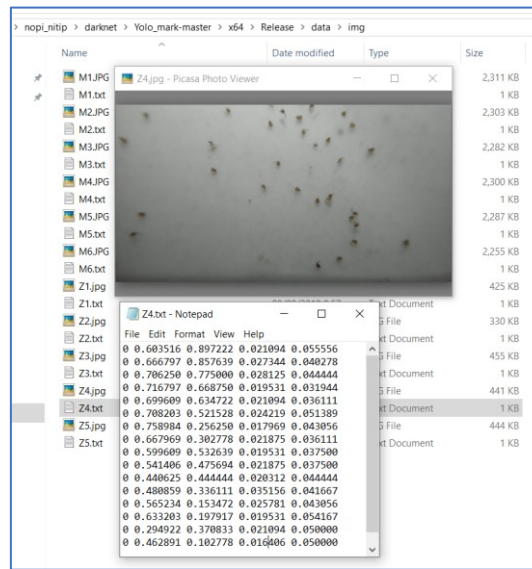
Gambar 29. Gambar yang akan dilatih

## 2) Pelabelan Gambar

Dari 200 gambar yang sudah disiapkan kemudian dilakukan proses anotasi citra. Pada proses anotasi ini tiap object larva pada gambar akan diberi kontak pembatas (bounding box) sesuai kategori kelasnya. Pada penelitian ini digunakan sistem operasi windows sehingga pada proses anotasi menggunakan tools yang bernama Yolo\_mark. Proses ini akan menghasilkan output berupa sebuah data yang berisikan informasi letak kotak batas beserta labelnya dalam bentuk file dengan ekstensi \*.txt.



Gambar 30. Prose Labeling menggunakan Yolo\_Mark



Gambar 31. Hasil Labelling

### 3) PreTraining

Sebelum dilakukan proses training, terlebih dahulu dilakukan persiapan sistem berupa beberapa aplikasi yang akan dipakai sebagai berikut:

#### a) File compile darknet

File darknet adalah program utama yang dijalankan dan YOLOv3 ada library yang digunakan. Untuk menggunakan algoritma YOLOv3 harus melalui file hasil compile yang diberi nama darknet oleh pengembangnya.

#### b) File cuDNN Nvidia CUDA

Proses pelatihan dengan menggunakan CPU memang cepat, namun dengan menggunakan GPU bisa 500 kali lebih cepat.

Untuk menggunakan pemrosesan dengan GPU, penelitian ini menggunakan file cuDNN Nvidia CUDA.

c) Google drive

Dikarenakan pemrosesan menggunakan layanan cloud computing berupa google colab, maka fasilitas berbagi media penyimpanan menggunakan goole drive.

d) Google Colab

Google colab merupakan sebuah fasilitas cloud computing yang disediakan oleh google. Google colab menyediakan sebuah computer virtual yang spesifikasi nya setara dengan sebuah computer jupiter. Penggunaan komputer virtual ini hanya diberi waktu maksimal 12 jam oleh google. Proses training data pada penelitian ini dilakukan di google colab.

4) Training

Pada tahapan ini, proses training menggunakan framework dari darknet. Pada framework ini digunakan model yang terdiri dari kode jaringan dan bobot pra-latih dari jaringan tersebut. File kode untuk model berupa file dengan ekstensi \*.cfg dan file pra-latih berupa file dengan ekstensi \*.weights. File konfigurasi atau file \*.cfg diatur terlebih dulu sebelum proses training. Pada file cfg perlu dilakukan beberapa pengaturan diantaranya

- a. Pengaturan class, terdapat 2 class yang akan digunakan yaitu class megalopa dan bukan megalopa

- b. Pengaturan jumlah filter layer, filter layer dihitung menggunakan rumus

$$B*(Class+5)$$

B = merupakan jumlah boundingbox untuk setiap cell penelitian ini menggunakan Yolo v3 yang mana memprediksi 3 boundingbox untuk setiap cell.

Class = terdapat 2 class

Sehingga filter layer =  $3*(2+5)$  diperoleh 21 jumlah filter layer.

Filter layer yang diatur pada layer terakhir pada model yang digunakan.

- c. Batas iterasi, pada umumnya batas iterasi dalam 1 class pada umumnya dilakukan 2000 iterasi. Batas iterasi pada konfigurasi ini digunakan 4000 iterasi.
- d. Pengaturan lain dapat dilihat pada gambar berikut

```
# Training
batch=64
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 4000
policy=steps
steps=3200,3600
scales=.1,.1
```

```
[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326
classes=2
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

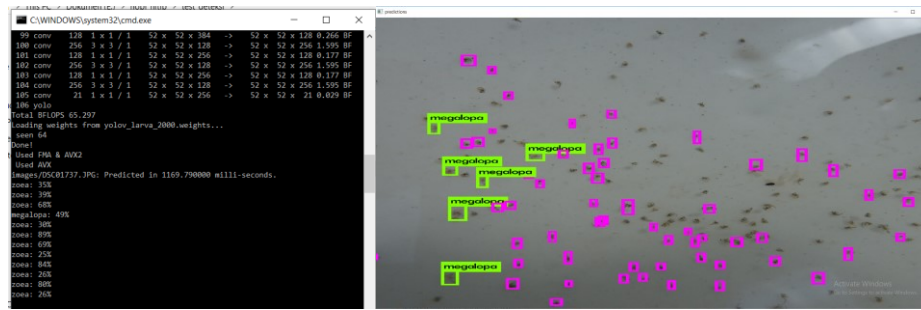
Gambar 32. Pengaturan file konfigurasi

Hasil dari proses pelatihan ini akan menghasilkan file bobot final dalam ekstensi \*.weights yang nantinya akan digunakan untuk melakukan transfer learning ketika akan melakukan proses deteksi.

## 5) Prediksi dan Deteksi

Proses prediksi dan deteksi dilakukan dengan memanfaatkan file kompilasi darknet, file kode model (\*.cfg) yang digunakan ketika proses training, dan file bobot final hasil pelatihan (\*.weight). Proses transfer learning dilakukan dengan mengeksekusi perintah yang memanfaatkan ketiga file tersebut dan file gambar yang akan dipakai untuk prediksi deteksi object megalopa dan bukan. Hasil dari eksekusi perintah adalah gambar yang sama dengan tambahan object yang sudah ditandai dalam bentuk kotak yang diberi nama sesuai kategorinya.





Gambar 33. Output Hasil Deteksi

#### 4. Pengiriman Notifikasi

Pada tahapan ini, jika hasil deteksi gambar pada komputer server terdapat larva yang telah memasuki fase megalopa, maka sistem akan mengirimkan informasi ke database gammu, yang selanjutnya akan diproses oleh library gammu untuk mengirimkan sms notifikasi ke petambak

#### E.Sumber Data

Tahapan pengumpulan data terdiri dari pengumpulan data primer dan pengumpulan data sekunder:

##### 1. Data Primer

Data primer yang digunakan dalam penelitian ini yaitu berupa gambar larva kepiting yang telah di ambil dari tambak penelitian kepiting. Gambar tersebut di ambil di kolam produksi benih kepiting BBAP Takalar, dengan menggunakan kamera webcam.

Data yang digunakan pada sistem ini terdiri atas 2 jenis data gambar yaitu:

##### 1) Data Latih

Data latih adalah data yang telah melalui proses latih sebelumnya. Data yang digunakan pada sistem ini sebanyak 200 sampel gambar dari 2 jenis fase larva kepiting (*zoea*, *megalopa*)

## 2) Data Uji

Data Uji merupakan data gambar yang diinput untuk selanjutnya menuju ke proses testing sistem. Selain melalui proses testing, data uji ini juga akan digunakan kembali sebagai bagian dari database sistem. Data gambar yang akan diuji berupa gambar larva kepiting dari *megalopa* dan bukan *megalopa*.

Pada penelitian ini, data direkam menggunakan kamera yang nantinya akan di control oleh raspberry pi dengan bahasa pemograman python.

## 2. Data Sekunder

Dalam metode ini dilakukan pencarian sebanyak mungkin literature yang ada, baik dari buku, jurnal maupun internet.

## F.Instrumen Penelitian

### 1. Software

- a. OS Windows 10 64-bit
- b. Pemrograman Python
- c. Darknet
- d. CUDA

### 2. Hardware

- a. Laptop dengan Prosesor Core i7 (2.60 GHz)

- b. Memory DDR3 16 GB
- c. Rasberry pi 3
- d. Acces Poin
- e. Kamera
- f. Modem

### G. Tolak Ukur Akurasi Sistem

Setelah dilakukan perancangan sistem dengan metode yang digunakan untuk deteksi larva, dilanjutkan dengan pengukuran kinerja sistem menggunakan confusion matrix. Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Nilai akurasi pengukuran kinerja menggunakan confusion matrix diperoleh dengan persamaan

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

Dimana :

- TP adalah True Positive, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem
- TN adalah True Negatif, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem
- FN adalah False Negatif yaitu jumlah data negatif namun terklasifikasi oleh sistem
- FP adalah false positive, yaitu jumlah data positif namun terklasifikasi salah oleh sistem

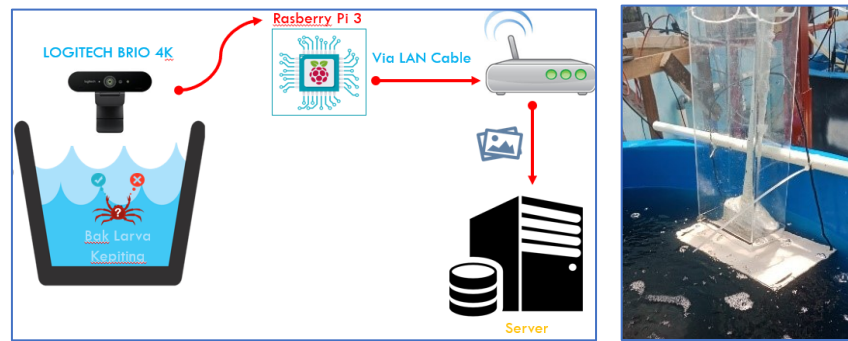
## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Akuisisi Data**

Akuisisi data merupakan proses perubahan data dari format awal menjadi format baru yang nantinya dapat diolah oleh sistem. Akuisisi data dalam penelitian ini berupa data gambar. Data gambar yang digunakan dalam penelitian adalah berupa gambar dengan ekstensi \*.jpg. Pengambilan data gambar menggunakan kamera webcam Logitech BRIO 4K Ultra HD.

Skenario pengambilan gambar dilakukan dengan meletakkan webcam pada posisi tenggelam dikedalaman 1 sampai 3 cm dari permukaan air, webcam dilindungi oleh acrylic untuk mencegah terkena air bak. Peletakan akriliki selain melindungi kamera terkena air juga dimaksudkan untuk mengurangi *noise* hasil pengambilan gambar akibat pantulan cahaya pada permukaan air, gelombang air, dan gangguan lain seperti proses aerasi yang terjadi dalam bak larva tersebut. Data gambar ini diambil langsung dilokasi penelitian yang bertempat di BPBAP Takalar



Gambar 34. Posisi Kamera pada Bak larva

### B. Analisa Sistem Deteksi Larva Kepiting Fase Megalopa

Sebagaimana dijelaskan pada BAB III, pada rancangan sistem deteksi larva dimana gambar diolah melalui tahapan preprocessing yaitu grayscale dan sharpening, kemudian hasil dari preprocessing di treshhold, setelah proses treshhold langkah selanjutnya dilakukan ekstraksi fitur dengan menentukan Region of Interest (ROI) untuk di cari nilai eigen valuenya hasil dari eigen value tersebut sebagai inputan pada proses klasifikasi menggunakan algoritma SVM. Dari sistem deteksi tersebut dilakukan pengujian pada 10 gambar data uji. Hasi pengujian sistem beserta perhitungan accuracy menggunakan metode penghitungan accuracy seperti yang telah di jelaskan pada bab III dapat dilihat pada Tabel 2 berikut ini

Tabel 2. Hasil pengujian sistem

No	Jumlah Megalopa		Confusion matrix				Accuracy
	Real	Sistem	TP	TN	FP	FN	
1	10	41	10	0	31	0	24%
2	21	2	2	0	0	19	10%

No	Jumlah Megalopa		Confusion matrix				Accuracy
	Real	Sistem	TP	TN	FP	FN	
3	2	29	2	0	27	0	7%
4	8	63	8	0	55	0	13%
5	9	39	9	0	30	0	23%
6	10	39	10	0	29	0	26%
7	9	35	9	0	26	0	26%
8	10	40	10	0	30	0	25%
9	5	32	5	0	27	0	16%
10	2	23	2	0	21	0	9%
<b>Rata-rata Accuracy</b>							<b>18 %</b>

Hasil akurasi dihitung menggunakan persamaan sebagaimana seperti yang di jelaskan pada BAB III. Hasil uji coba menggunakan sistem deteksi tersebut diperoleh rata-rata akurasi dengan nilai 18%. Rata-rata akurasi yang rendah terjadi karena sistem deteksi tersebut masih banyak terdapat kesalahan deteksi larva dimana larva fase non megalopa yang terdeteksi sebagai megalopa atau sebaliknya. Dengan hasil yang demikian maka sistem deteksi menggunakan sistem tersebut belum dapat diterapkan.

Sistem deteksi dirancang kembali menggunakan YOLOv3 sebagaimana di jelaskan pada BAB III. Sistem yang sudah di rancang kemudian di ujikan. Tabel xx berikut merupakan tabel hasil pengujian sistem deteksi menggunakan YOLOv3 dimana digunakan gambar uji yang pada keseluruhan objek merupakan objek non megalopa.

Tabel 3 berikut menunjukkan hasil deteksi dari sistem sistem deteksi menggunakan YOLOv3 kemudian dibandingkan dengan hitungan real secara manual dari 10 data uji terdiri dari gambar megalopa dan bukan megalopa diperoleh data sebagai berikut:

Tabel 3. Proses uji dengan gambar megalopa dan bukan megalopa

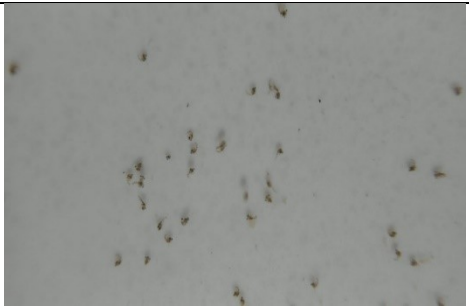
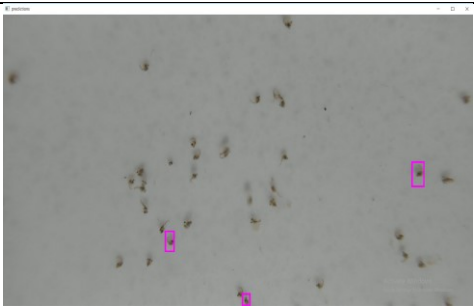


No	Jumlah Megalopa		Confusion matrix				Akurasi (%)
	Real	Sistem	TP	TN	FP	FN	
1	4	3	4	0	0	1	75.00
2	13	11	13	0	0	2	84.62
3	2	2	2	0	0	0	100.00
4	3	3	3	0	0	0	100.00
5	11	9	11	0	0	2	81.82
6	5	4	5	0	0	1	80.00
7	12	11	12	0	0	1	91.67
8	9	8	9	0	0	1	88.89
9	12	11	12	0	0	1	91.67
10	6	5	6	0	0	1	83.33
Rata-rata Akurasi							87.7%

Pada data nomer 1 terdapat gambar dimana pada gambar tersebut ketika dihitung secara manual terdapat 4 megalopa akan tetapi disistem terdeteksi 3 megalopa dari hasil tersebut dengan menggunakan persamaan akurasi

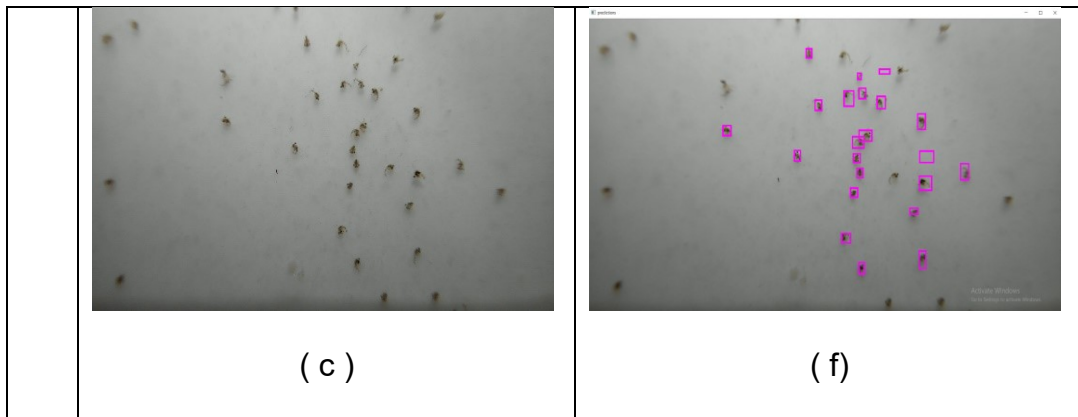
sebagaimana dijelaskan pada BAB III diperoleh hasil akurasiya yaitu  $((3+0)/(3+0+1+0) \times 100\%$  sehingga diperoleh 75%. Begitu juga untuk gambar 2 hingga gambar 10 yang kemudian hasil akurasi tersebut dihitung rata-ratanya dan rata-rata akurasi yang diperoleh yaitu 88%.

Sistem deteksi menggunakan Yolov3 kemudian di ujikan kembali menggunakan gambar uji dimana pada gambar yang di ujikan keseluruhannya merupakan objek non megalopa. Tabel 4 berikut ini menunjukkan hasil pengujian gambarnya

Tabel 4. Gambar uji non megalopa


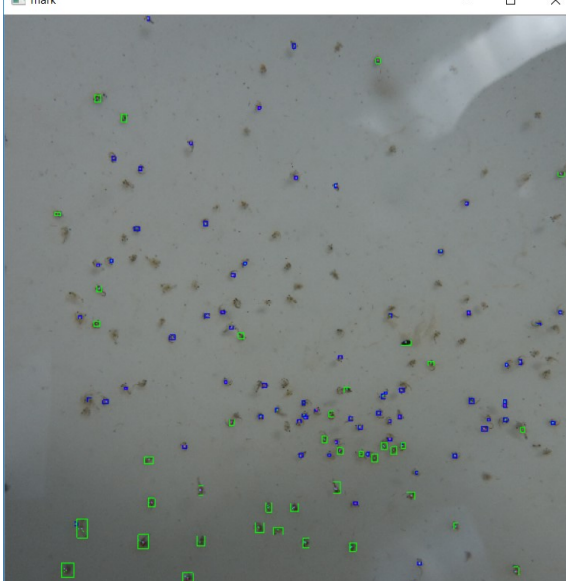
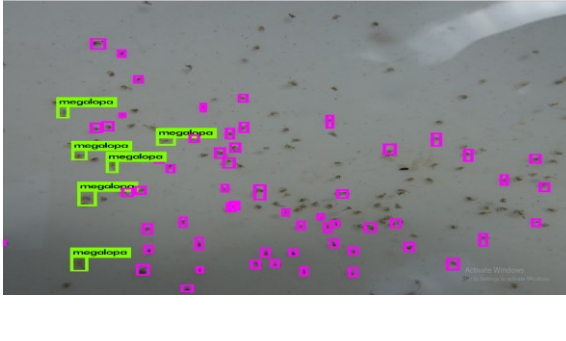
No	Gambar Uji	Gambar hasil deteksi
	 <p data-bbox="597 1318 643 1360">(a)</p>	 <p data-bbox="1097 1318 1143 1360">(d)</p>
	 <p data-bbox="597 1730 643 1772">(b)</p>	 <p data-bbox="1097 1730 1143 1772">(e)</p>





Dari hasil pengujian gambar tersebut sistem mampu mendeteksi hasil gambar dimana pada seluruh gambar bukan megalopa pada sistem ini juga tidak terdeteksi adanya megalopa.

Kemudian pada peneletian ini di uji cobakan gambar untuk dideteksi salah satu gambar perbandingan pengujian gambar menggunakan algoritma svm dan algoritma YOLOv3 menggunakan gambar yang sama diperoleh hasil sebagai berikut

	<p>Gambar 35. Gambar uji</p> <p>Merupakan gambar asli jika dihitung secara manual terdapat 8 megalopa</p>
	<p>Gambar 36. Hasil deteksi</p> <p>Deteksi menggunakan proses grayscale, sharpening, thresholding, dan klasifikasi menggunakan algoritma svm diperoleh Jumlah Megalopa sebanyak 39</p>
	<p>Gambar 37. Hasil deteksi menggunakan YOLOV3</p> <p>Deteksi menggunakan Algoritma YOLOv3 diperoleh Jumlah Megalopa sebanyak 6</p>

Pada gambar 36 dan gambar 37 kedua sistem diatas mampu untuk mendeteksi fase megalopa. Akurasi jumlah megalopa yang terdeteksi menggunakan algoritma YOLOv3 mendekati jumlah yang sesungguhnya. Sehingga pada penelitian ini sistem deteksi yang diterapkan menggunakan sistem deteksi menggunakan algoritma YOLOv3.

### C.Analisis Kerja Sistem

#### a. Pengambilan Gambar

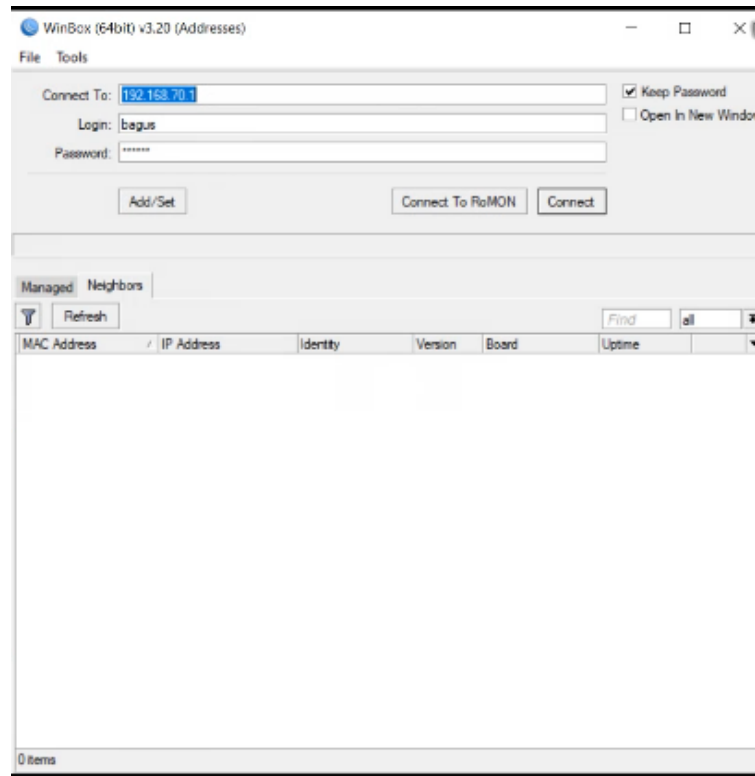
Pada proses pengambilan gambar, server akan memberikan perintah pada raspberry pi. Sehingga perlu dipastikan pengaturan setting dimana server dapat berkomunikasi dengan raspberry pi dalam hal ini baik server maupun raspberry pi terkoneksi menggunakan mikrotik.

1. Server dihubungkan dengan wifi mikrotik, begitu juga dengan raspberry pi



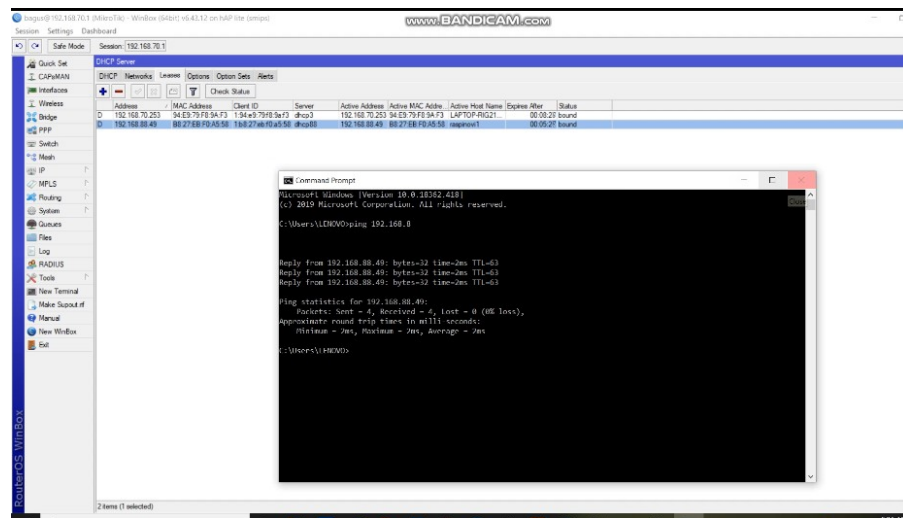
Gambar 38. Pengaturan koneksi

## 2. Kemudian proses menghubungkan raspberry pi



Gambar 39. Pengaturan koneksi raspberi pi

3. Setelah terkoneksi baik raspberry pi maupun server memiliki ip masing-masing, kemudian dilakukan pengetesan dengan pemanggilan alamat ip baik disisi server maupun disisi raspberry pi. Seperti pada gambar berikut baik disisi server maupun disisi raspberry pi telah saling reply sehingga dapat dipastikan kedua perangkat telah saling terkoneksi

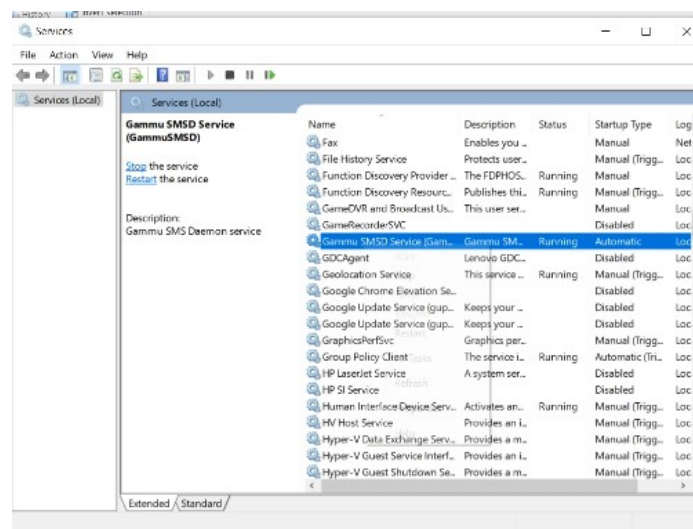


Gambar 40. Pengujian koneksi raspberry pi dengan server

## b. Pengaturan Gammu

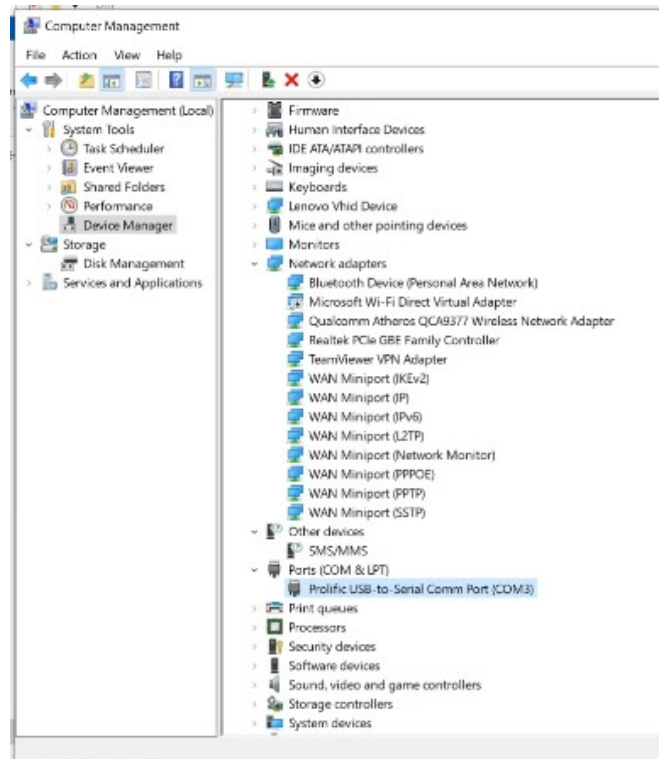
Gammu digunakan untuk menjembatani komunikasi dalam hal pengiriman sms

### 1. Mengaktifkan layanan gammu



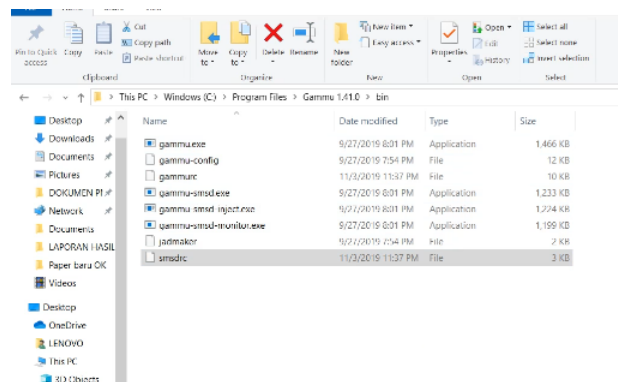
Gambar 41. Pengaktifan layanan gammu

## 2. Melihat port yang aktif pada gammu



Gambar 42. Pengecekan port gammu

## 3. Memastikan pada file gammurc dan smsdrc sudah disetting pada gammu yang aktif



Gambar 43. Pengaturan settingan port pada gammurc dan smsdrc

Pada langkah sebelumnya diketahui bahwa gammu terhubung pada port 3 maka pengaturan di kedua file baik gammurc dan smsrc juga harus tersetting pada port 3 seperti pada gambar berikut

```

1 : This is a sample ~/.gammurc file.
2 : In this/linux copy it into your home directory and name it .gammurc
3 :   or into /etc and name it gammurc
4 : In Win32 copy it into directory with Gammu.exe and name gammurc
5 : More about parameters later
6 : Anything behind ; or # is comment.
7 :
8 : -----
9 [gammu]
10
11 device = com3:
12 connection = at115200
13 ; DO NOT use model configuration unless you really need it
14 model = f110
15 ;synchronous_time = yes
16 logfile = C:\Users\LENOVO\AppData\Local\Gammu\gammulog
17 logformat = textall
18 ;use_locking = yes
19 ;gammuloc = locfile
20 ;startinfo = yes
21 ;gammucoding = utf8
22 ;usephonedb = yes
23
24 [gammu1]
25
26 ;device = com3:
27 ;model = f110
28 ;connection = rs485bus
29 ;synchronous_time = yes
30 ;logfile = gammulog
31 ;logformat = textall
32 ;use_locking = yes
33 ;gammuloc = locfile
34 ;startinfo = yes
35 ;gammucoding = utf8
36
37 ; Step 1. Please find required Connection parameter and look into assigned
38 ; with it device type. With some Connection you must set concrete model
39
40 ; ----- cables -----
41 ; New Sokie protocol for FDU0/UM09
42 ; Connection "Ebus", Device type serial
43 ; New Sokie protocol for DLR3/UM83
44 ; Connection "Ebusdir"/"3lr3", Device type serial

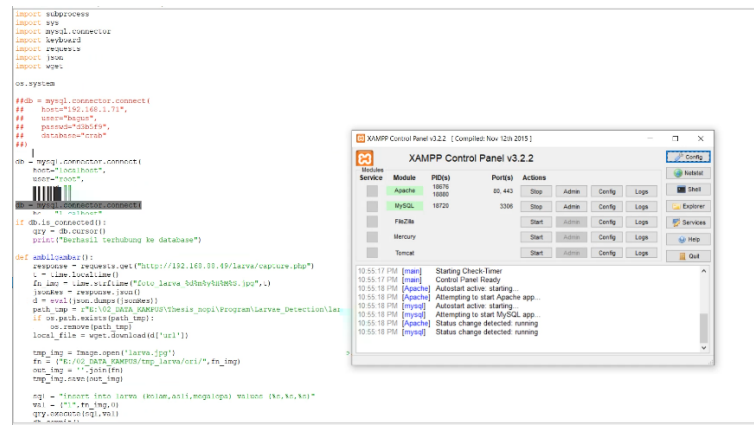
```

Gambar 44. Pengaturan gammu pada file gammurc dan smsrc

### c. Menjalankan program deteksi

Setelah server dan raspberry pi saling terkoneksi kemudian juga sudah di lakukan pengaturan Gammu maka langkah berikutnya yaitu menjalankan program

1. Pastikan database pada server sudah di aktifkan . Database ini akan menyimpan seluruh gambar yang di capture , gambar setelah di lakukan deteksi serta akan menyimpan data pengiriman notifikasi/sms. Berikut gambar mengaktifkan database



Gambar 45. Pengaktifan layanan database

## 2. Memastikan alamat ip raspberry pi pada program sudah sesuai

```
File Edit Format Run Options Window Help
import subprocess
import sys
import mysql.connector
import keyboard
import requests
import json
import wget

os.system

##db = mysql.connector.connect(
## host="192.168.1.71",
## user="bagus",
## passwd="d3b5f9",
## database="crab"
##)

db = mysql.connector.connect(
host="localhost",
user="root",
passwd="",
database="crab"
)

if db.is_connected():
    qry = db.cursor()
    print('Berhasil terhubung ke database')

def ambilgambar():
    response = requests.get("http://192.168.1.71/larva/capture.php")
    l = json.loads(response.json())
    fn_img = l['file']
    jsonkes = response.json()
    d = json.loads(jsonkes)
    path_tmp = r'E:\02_DATA_KAMPUS\thesis_nopi\Program\larva_Detection\larva.jpg'
    if os.path.exists(path_tmp):
        os.remove(path_tmp)
    local_file = wget.download(d['url'])

    tmp_img = image.open('larva.jpg')
    fn = (r'E:\02_DATA_KAMPUS\tmp_larva\ori/',fn_img)
    out_img = ''.join(fn)
    tmp_img.save(out_img)

    qry = "insert into larva (kolam,asi,megalopa) values (%s,%s,%s)"
    val = (l['k'],fn_img,0)
    qry.execute(qry, val)
    db.commit()
```

Gambar 46. Pengecekan alamat ip raspberry pi

## 3. Memastikan kembali folder penyimpanan gambar hasil capture dan hasil



```

File Edit Format Run Options Window Help
import json
import wget

os.system

##db = mysql.connector.connect(
##     host="192.168.1.71",
##     user="bacus",
##     password="d3b5f9",
##     database="crab"
##)

db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="crab"
)

if db.is_connected():
    qry = db.cursor()
    print("Berhasil terhubung ke database")

def ambilgambar():
    response = requests.get("http://192.168.88.49/larva/capture.php")
    t = time.localtime()
    fn_img = time.strftime("foto_larva_%d%my%M%S.jpg", t)
    jsonRes = response.json()
    d = eval(json.dumps(jsonRes))
    path_tmp = "E:/02_DATA_KAMPUS/Thesis_nopi/Program/Larvae_Detection/larva.jpg"
    if os.path.exists(path_tmp):
        os.remove(path_tmp)
    local_file = wget.download(d['url'])

    tmp_img = image.open('larva.jpg')
    fn = ("E:/02_DATA_KAMPUS/tmp_larva/crabi/",fn_img)
    out_img = ''.join(fn)
    tmp_img.save(out_img)

    sql = "insert into larva (kolam,asli,megalopa) values (%s,%s,%s)"
    val = ('1',fn_img,0)
    qry.execute(sql,val)
    db.commit()

    print("ambil image berhasil")

def deteksi():
    path = "E:/02_DATA_KAMPUS/tmp_larva/deteksi/"
    sql = "select id,asli from larva where flag = 0"
    qry.execute(sql)
    res = qry.fetchall()
    for x in res:
        ....

```

Gambar 47. Pengaturan alamat capture

```

File Edit Format Run Options Window Help
##db = mysql.connector.connect(
##     host="192.168.1.71",
##     user="bacus",
##     password="d3b5f9",
##     database="crab"
##)

db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="crab"
)

if db.is_connected():
    qry = db.cursor()
    print("Berhasil terhubung ke database")

def ambilgambar():
    response = requests.get("http://192.168.88.49/larva/capture.php")
    t = time.localtime()
    fn_img = time.strftime("foto_larva_%d%my%M%S.jpg", t)
    jsonRes = response.json()
    d = eval(json.dumps(jsonRes))
    path_tmp = "E:/02_DATA_KAMPUS/Thesis_nopi/Program/Larvae_Detection/larva.jpg"
    if os.path.exists(path_tmp):
        os.remove(path_tmp)
    local_file = wget.download(d['url'])

    tmp_img = image.open('larva.jpg')
    fn = ("E:/02_DATA_KAMPUS/tmp_larva/crabi/",fn_img)
    out_img = ''.join(fn)
    tmp_img.save(out_img)

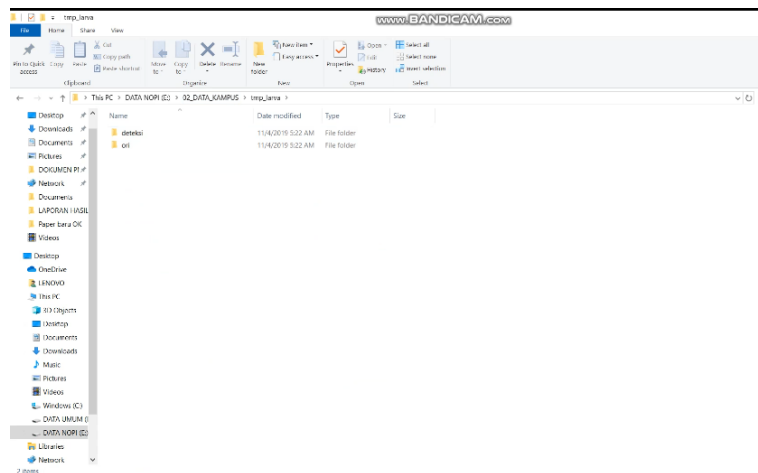
    sql = "insert into larva (kolam,asli,megalopa) values (%s,%s,%s)"
    val = ('1',fn_img,0)
    qry.execute(sql,val)
    db.commit()

    print("ambil image berhasil")

def deteksi():
    path = "E:/02_DATA_KAMPUS/tmp_larva/deteksi/"
    sql = "select id,asli from larva where flag = 0"
    qry.execute(sql)
    res = qry.fetchall()
    for x in res:
        ....

```

Gambar 48. Pengaturan alamat output



Gambar 49. Lokasi penyimpanan file capture dan output

4. Pengaturan durasi waktu pengambilan gambar. Pada penelitian ini dibuat 10 detik

```

### = time.strftime("%d-%m-%y %H:%M:%S",t)

output_img = ''.join(path.in_img)
screenshot_image.save(output_img)

val = "update larva set deteksi = %s,mealupa = %s,flag = 1 where id = %s"
val = (fn_img,num,a)
qry.execute(sql,val)

m = ''.join("pada ",fn_img," ada ",str(numM)," mealupa yang terdeteksi")
n = ''.join("pada kolom 1, ada ",str(numM)," mealupa yang terdeteksi pada ",sms)
print(m)

if xAMM > 0:
    sql = "insert into outbox (DestinationNumber,TextDecodeded,CreatorID) values (%s,%s,%s)"
    val = ("%0123456789",s,"Movel")
    qry.execute(sql,val)

def loopimg():
    while True:
        try:
            t = time.localtime()
            #current_time = time.strftime("%d-%m-%y %H:%M:%S",t)
            current_time = time.strftime("%d-%m-%y %H:%M:%S",t)
            print("Current time = ", current_time)
            ambilgambar()
            deteksi()
            #ambilgambar()
            #qry.execute(sql,cmd)
            break

loopimg()

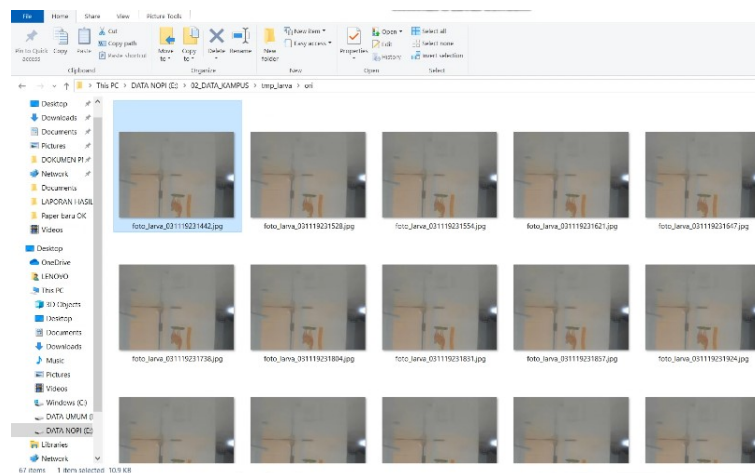
#qry = db.cursor()
#cmd =
#qry.execute("CREATE DATABASE toke_maihas")
#id = 1

#path to tar file
#tar_path = r'E:\nopi_nitip\larvae_detection\imgpack.tar'
#tarbz_name = (tar_path,".tar") as tarbz
## for line in tarbz:
##     image_list = [num for num in line.split()] for line in tarbz
##for image in image_list:
##     command = ["@remot.exe detector test data/obj.data.ctg/yolow_larwa.ctg yolow_larwa_2000.weights -dont_show", image()]
##     command = ["@remot.exe", "detector", "test", "data/obj.data", "ctg/yolow_larwa.ctg", "yolow_larwa_2000.weights", "-dont_show", image()]

```

Gambar 50. Pengaturan durasi waktu capture image

5. Jalankan program setelah melakukan pengaturan di program  
Berikut tampilan hasil capture



Gambar 51. Hasil capture image

Berikut hasil proses deteksi dari gambar yang telah di ambil sebelumnya

```

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (tags/v3.7.3:10444ad2, Mar 26 2019, 21:26:13) [AMD64] v.1916.302.0611 (Python
1) on win32
Type "help()" for more
>>> C:\WINDOWS\system32\cmd.exe
>>>
47 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
48 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
49 Shortcut Layer: 46
50 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
51 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
52 Shortcut Layer: 49
53 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
54 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 DF
55 Shortcut Layer: 52
56 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 DF
57 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 DF
58 Shortcut Layer: 55
59 conv 256 1 x 1 / 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 DF
60 conv 512 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 DF
61 Shortcut Layer: 58
62 conv 1024 3 x 3 / 2 26 x 26 x 512 -> 13 x 13 x1024 1.595 BF
63 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
64 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
65 Shortcut Layer: 62
66 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
67 conv 1024 3 x 3 / 1 13 x 13 x 512 -> 13 x 13 x1024 1.595 BF
68 Shortcut Layer: 65
69 conv 512 1 x 1 / 1 14 x 14 x1024 -> 14 x 14 x 512 0.177 BF
70 conv 1024 3 x 3 / 1 14 x 14 x 512 -> 14 x 14 x1024 1.595 BF
71 Shortcut Layer: 68
72 conv 512 1 x 1 / 1 14 x 14 x1024 -> 14 x 14 x 512 0.177 BF
73 conv 1024 3 x 3 / 1 14 x 14 x 512 -> 14 x 14 x1024 1.595 BF
74 Shortcut Layer: 71
75 conv 512 1 x 1 / 1 13 x 13 x1024 -> 13 x 13 x 512 0.177 BF
76

```

Gambar 52. Proses deteksi hasil capture image larva

Berikut gambar tabel pada mysql yang berisi gambar hasil capture dan gambar hasil deteksi



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A.Kesimpulan**

Berdasarkan hasil penelitian dan pengujian sistem deteksi larva kepiting fase megalopa menggunakan algoritma CNN pada YOLOv3 mampu mendeteksi fase megalopa berdasarkan ukuran larva. Uji coba sistem dilakukan menggunakan beberapa gambar. Pada ujicoba dengan menggunakan gambar uji dimana gambar terdiri dari larva kepiting bukan fase megalopa sistem menunjukkan hasil bahwa tidak ada megalopa yang terdeteksi. Pada ujicoba menggunakan gambar dimana gambar terdiri fase megalopa dan bukan megalopa sistem mampu mendeteksi larva kepiting fase megalopa dengan akurasi mencapai 87.7%. Dengan hasil dimana akurasi mencapai 87.7% menunjukkan bahwa penerapan teknologi pengolahan citra untuk deteksi fase megalopa dapat diterapkan untuk mendeteksi fase megalopa pada larva kepiting.

#### **B.Saran**

Adapun saran yang ingin penulis sampaikan untuk penelitian selanjutnya:

1. Pada tahap pengambilan gambar, gambar larva kepiting yang dapat di capture menggunakan kamera yang dilakukan saat penelitian memiliki luasan area yang terbatas dikarenakan objek yang akan diambil

memiliki ukuran yang relatif kecil. Peneliti selanjutnya dapat memilih kamera yang lebih sesuai untuk mendapat tangkapan luas area yang lebih luas dengan pertimbangan objek yang relatif kecil dengan pergerakan yang relatif cepat.

2. Sistem deteksi larva kepiting fase megalopa dapat diintegrasikan dengan sistem otomatisasi lainnya seperti otomatisasi penambahan pemberian pakan atau otomatisasi pemisahan larva sehingga dapat menekan kanibalisme larva yang pada akhirnya dapat meningkatkan survival rate pada budidaya larva kepiting

## DAFTAR PUSTAKA

- A, R.A., Sari, I.N., Arinda, V.I., 2017. RANCANG BANGUN PENGHITUNG BENIH IKAN MENGGUNAKAN BINARY THRESHOLDING PADA RASPBERRY PI SECARA REAL TIME. JIP 4, 1. <https://doi.org/10.33795/jip.v4i1.137>
- Albawi, S., Mohammed, T.A., Al-Zawi, S., n.d. Understanding of a Convolutional Neural Network 7.
- Arymurthy, A.M., 1992. Pengantar Pengolahan Citra. Jakarta: Elex Media Komputindo.
- Bernilai Ekonomi Tinggi, KKP Benihkan Rajungan secara Massal di Takalar | MaritimNews.com [WWW Document], n.d. URL <http://maritimnews.com/2017/06/bernilai-ekonomi-tinggi-kkp-benihkan-rajungan-secara-massal-di-takalar/> (accessed 10.5.19).
- Bruun, J.M., Kapel, C.M.O., Carstensen, J.M., 2012. Detection and classification of parasite eggs for use in helminthic therapy, in: 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI). Presented at the 2012 IEEE 9th International Symposium on Biomedical Imaging (ISBI 2012), IEEE, Barcelona, Spain, pp. 1627–1630. <https://doi.org/10.1109/ISBI.2012.6235888>
- Darsono, P., 1997. MATING BEHAVIOR OF THE BLUE CRAB, CALUNECTES SAPIDUS RATHBUN 10.
- Djunaedi, A., 2009. Kelulushidupan dan Pertumbuhan Crablet Rajungan (*Portunus pelagicus* Linn.) Pada Budidaya dengan Substrat Dasar yang Berbeda 14, 4.
- Flores, A., Crisostomo, P., Lopez, J., 2008. Peruvian Scallop larvae counting system using image processing techniques, in: 2008 7th International Caribbean Conference on Devices, Circuits and Systems. Presented at the 2008 7th International Caribbean Conference on Devices, Circuits and Systems (ICCDACS), IEEE, Cancun, Mexico, pp. 1–4. <https://doi.org/10.1109/ICCDACS.2008.4542660>

- Guo, J., Zhao, S., Shu, M., Yan, Z., Sun, M., Zhao, X., Feng, X., 2014. Trajectory tracking of spasm-oriented zebrafish larvae, in: Proceedings of the 33rd Chinese Control Conference. Presented at the 2014 33rd Chinese Control Conference (CCC), IEEE, Nanjing, China, pp. 6849–6853. <https://doi.org/10.1109/ChiCC.2014.6896128>
- Harto, D., Sarosa, M., 2012. Aplikasi Jaringan Syaraf Tiruan untuk Mendeteksi White Spot 6, 8.
- Jose, J., n.d. Seed production and farming of blue swimmer crab *Portunus pelagicus* 8.
- Juwana, S., 1997. TINJAUAN TENTANG PERKEMBANGAN PENELITIAN BUDIDAYA RAJUNGAN (*PORTUNUS PELAGICUS*) 12.
- Lainez, S.M.D., Gonzales, D.B., 2019a. Automated Fingerlings Counting Using Convolutional Neural Network, in: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). Presented at the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), IEEE, Singapore, pp. 67–72. <https://doi.org/10.1109/CCOMS.2019.8821746>
- Lainez, S.M.D., Gonzales, D.B., 2019b. Automated Fingerlings Counting Using Convolutional Neural Network, in: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). Presented at the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), IEEE, Singapore, pp. 67–72. <https://doi.org/10.1109/CCOMS.2019.8821746>
- Li, G., Song, Z., Fu, Q., 2018. A New Method of Image Detection for Small Datasets under the Framework of YOLO Network, in: 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Presented at the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), IEEE, Chongqing, China, pp. 1031–1035. <https://doi.org/10.1109/IAEAC.2018.8577214>
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition



(CVPR), IEEE, Las Vegas, NV, USA, pp. 779–788.  
<https://doi.org/10.1109/CVPR.2016.91>

Redmon, J., Farhadi, A., n.d. YOLOv3: An Incremental Improvement 6.

Romano, N., Zeng, C., 2017. Cannibalism of Decapod Crustaceans and Implications for Their Aquaculture: A Review of its Prevalence, Influencing Factors, and Mitigating Methods. *Reviews in Fisheries Science & Aquaculture* 25, 42–69.  
<https://doi.org/10.1080/23308249.2016.1221379>

Yang, W., Jiachun, Z., 2018. Real-time face detection based on YOLO, in: 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII). Presented at the 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), IEEE, Jeju, pp. 221–224. <https://doi.org/10.1109/ICKII.2018.8569109>

Zheng, X., Cai, R., 2016. Geometrid larvae detection using contour feature, in: 2016 IEEE 13th International Conference on Signal Processing (ICSP). Presented at the 2016 IEEE 13th International Conference on Signal Processing (ICSP), IEEE, Chengdu, China, pp. 673–676.  
<https://doi.org/10.1109/ICSP.2016.7877917>

Zmora, O., Findiesen, A., Stubblefield, J., Frenkel, V., Zohar, Y., 2005. Large-scale juvenile production of the blue crab *Callinectes sapidus*. *Aquaculture* 244, 129–139.  
<https://doi.org/10.1016/j.aquaculture.2004.11.012>

## DAFTAR LAMPIRAN

1. Kode program sistem deteksi menggunakan Preprocessing, Morphological operasional dan klasifikasi menggunakan *Super Vector Machine (SVM)*

```

import cv2

import numpy as np

from math import atan2, cos, sin, sqrt, pi

eigenvalues_feature=np.array([])

eigenvectors_feature=np.array([])

def loadTrainingDataEigenVectors():

    #hasil ekstraksi eigenvector dari data training megalopa dan selain
megalopa

    labels = np.array([1, -1, -1, 1])

    trainingData = np.matrix([[0.2151,0.13771,0.5125,-0.3232], [0.1724,-
0.2215,-0.21322,0.1322],          [-0.13122,-0.26123,0.4121,0.2013],
[0.1322,0.3514,0.6134,-0.1312]], dtype=np.float32)

    return labels, trainingData

def loadTrainingDataEigenValues():

    #hasil ekstraksi eigenvalue dari data training megalopa dan selain
megalopa

    labels = np.array([1, -1, -1, 1])

    trainingData = np.matrix([[157, 35], [46, 17], [98, 35], [178, 110]],
dtype=np.float32)

```

```

    return labels, trainingData

def classificationModel(labels,trainingData):

    svm = cv2.ml.SVM_create()

    svm.setType(cv2.ml.SVM_C_SVC)

    svm.setKernel(cv2.ml.SVM_LINEAR)

    svm.setTermCriteria((cv2.TERM_CRITERIA_MAX_ITER, 100, 1e-6))

    svm.train(trainingData, cv2.ml.ROW_SAMPLE, labels)

    return svm

def classificationResult(model,testData):

    sampleMat = np.matrix([[testData[0,0],testData[1,0]]], dtype=np.float32)

    predict = model.predict(sampleMat)[1]

    return predict

def drawAxis(img, p_, q_, colour, scale):

    p = list(p_)

    q = list(q_)

    angle = atan2(p[1] - q[1], p[0] - q[0]) # angle in radians

    hypotenuse = sqrt((p[1] - q[1]) * (p[1] - q[1]) + (p[0] - q[0]) * (p[0] - q[0]))

    # Here we lengthen the arrow by a factor of scale

    q[0] = p[0] - scale * hypotenuse * cos(angle)

    q[1] = p[1] - scale * hypotenuse * sin(angle)

    cv2.line(img, (int(p[0]), int(p[1])), (int(q[0]), int(q[1])), colour, 1,

cv2.LINE_AA)

    # create the arrow hooks

```

```

p[0] = q[0] + 9 * cos(angle + pi / 4)
p[1] = q[1] + 9 * sin(angle + pi / 4)
cv2.line(img, (int(p[0]), int(p[1])), (int(q[0]), int(q[1])), colour, 1,
cv2.LINE_AA)

p[0] = q[0] + 9 * cos(angle - pi / 4)
p[1] = q[1] + 9 * sin(angle - pi / 4)
cv2.line(img, (int(p[0]), int(p[1])), (int(q[0]), int(q[1])), colour, 1,
cv2.LINE_AA)

def getOrientation(pts, img, model):
    sz = len(pts)
    data_pts = np.empty((sz, 2), dtype=np.float64)
    for i in range(data_pts.shape[0]):
        data_pts[i,0] = pts[i,0,0]
        data_pts[i,1] = pts[i,0,1]
    # Perform PCA analysis
    mean = np.empty((0))
    mean, eigenvectors, eigenvalues = cv2.PCACompute2(data_pts, mean)
    # Store the center of the object
    cntr = (int(mean[0,0]), int(mean[0,1]))
    cv2.circle(img, cntr, 3, (255, 0, 255), 2)
    p1 = (cntr[0] + 0.02 * eigenvectors[0,0] * eigenvalues[0,0], cntr[1] + 0.02
* eigenvectors[0,1] * eigenvalues[0,0])

```

```

    p2 = (cntr[0] - 0.02 * eigenvectors[1,0] * eigenvalues[1,0], cntr[1] - 0.02
* eigenvectors[1,1] * eigenvalues[1,0])
    drawAxis(img, cntr, p1, (0, 255, 0), 1)
    drawAxis(img, cntr, p2, (255, 255, 0), 5)
    angle = atan2(eigenvectors[0,1], eigenvectors[0,0]) # orientation in
radians

    #prediction using eigenvectors
    #predict=classificationResult(model,eigenvectors)

    #prediction using eigenvalues
    predict=classificationResult(model,eigenvalues)

    #return angle
    return predict

image = cv2.imread('E:\Gambar_Uji_SVM\svm_10.JPG') #ini ganti
gambare sesuaikan sama data yg mau diprediksi
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
cv2.namedWindow('grayscale',cv2.WINDOW_NORMAL)
cv2.resizeWindow('grayscale', 600,600)
cv2.imshow('grayscale', gray)
cv2.waitKey(0)

#Sharpen

#Blur method

blur = cv2.GaussianBlur(gray,(0,0) ,2)

#blur = cv2.medianBlur(gray,23)

```

```
#blur = cv2.bilateralFilter(gray,15,150,110)

#end blur method

#add blur to real image

gray1 = cv2.addWeighted(blur,0.12,gray,1.30,0)

#do twice, for more sharp image

#kondisi Gelap

gray = cv2.addWeighted(gray1,0.41,blur,0.43,0)

#gray = cv2.addWeighted(gray1,0.32,blur,0.55,0)

#kondisi Terang

#gray = cv2.addWeighted(gray1,0.43,blur,0.10,0)

#end add blur to real image

cv2.namedWindow('Sharpen',cv2.WINDOW_NORMAL)

cv2.resizeWindow('Sharpen', 600,600)

cv2.imshow('Sharpen', gray)

cv2.waitKey(0)

#binary

ret,thresh = cv2.threshold(gray,80,255,0)

cv2.namedWindow('threshold',cv2.WINDOW_NORMAL)

cv2.resizeWindow('threshold', 600,600)

cv2.imshow('threshold', thresh)

cv2.waitKey(0)

#dilation

kernel = np.ones([25,30], np.uint8)
```

```
img_dilation = cv2.dilate(thresh, kernel, iterations=1)
cv2.namedWindow('dilatation',cv2.WINDOW_NORMAL)
cv2.resizeWindow('dilatation', 600,600)
cv2.imshow('dilatation', img_dilation)
cv2.waitKey(0)

#ctrs, hier = cv2.findContours(thresh.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

#sort contours

#sorted_ctrs = sorted(ctrs, key=lambda ctr: cv2.boundingRect(ctr)[0])
contours, hier= cv2.findContours(thresh.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
sorted_ctrs = sorted(contours, key=lambda ctr: cv2.boundingRect(ctr)[0])
megalopaCount=0
zoeaCount=0
rectVal=0

#use eigenvectors as feature
#loadTrainingDataEigenVectors()

#use eigenvalues as feature
labels, trainingData=loadTrainingDataEigenValues()
model=classificationModel(labels,trainingData)

for i, ctr in enumerate(sorted_ctrs):

    # Calculate the area of each contour
    area = cv2.contourArea(ctr)
```

```
# Ignore contours that are too small or too large
if area < 1e2 or 1e5 < area:
    continue

# Find the orientation of each shape
# Get bounding box
x, y, w, h = cv2.boundingRect(ctr)

# Getting ROI
roi = image[y:y+h, x:x+w]

# show ROI
#cv2.imshow('segment no:'+str(i),roi)

if w > 15 and h > 15:
    orientationval= getOrientation(ctr, image,model)

    if orientationval==1:
        cv2.rectangle(image,(x,y),( x + w, y + h ),(0,255,0),4)
        #print(str(orientationval))
        megalopaCount=megalopaCount+1
    else:
        cv2.rectangle(image,(x,y),( x + w, y + h ),(255,0,0),4)
        #print(str(orientationval))
        zoeaCount=zoeaCount+1

#cv2.waitKey(0)

#if w > 15 and h > 15:
    #cv2.imwrite('D:\\}.png'.format(i), roi)
```



```

rectVal=zoeaCount+megalopaCount
print("Jumlah Larva : "+str(rectVal))
print("Jumlah Megalopa : "+str(megalopaCount))
print("Jumlah Zoea : "+str(zoeaCount))
cv2.namedWindow('mark',cv2.WINDOW_NORMAL)
cv2.resizeWindow('mark', 600,600)
cv2.imshow('mark',image)
cv2.waitKey(0)

```

## 2. Kode Program Sistem Deteksi Menggunakan Yolov3

### a. Capture Gambar

#### 1. File Capture.py

```

import os

import datetime

import sys

import time

import subprocess

script_dir = os.path.dirname(__file__)

currentdate = datetime.datetime.now().strftime("%Y-%m-
%d_%H%M%S")

real_path = "larva.jpg"

abs_file_path = os.path.join(script_dir,real_path)

print abs_file_path

```

```
os.system('fswebcam -r 4096x2160 -S 30 --no-banner ' +
abs_file_path)
```

## 2. Capture.php

```
<?php
$output = shell_exec('python /var/www/html/larva/cap3.py');
$msg=
array("path"=>$output,"fn"=>trim(basename($output)),"url"=>trim
('http://'.$_SERVER['SERVER_ADDR'].'/larva/'.basename($output)
));
echo json_encode($msg);
?>
```

## b. Kode Program Deteksi

```
import os
import glob
import time
import PIL
import PIL.Image as Image
import subprocess
import sys
import mysql.connector
import keyboard
import requests
import json
```

```
import wget

os.system

##db = mysql.connector.connect(
##  host="192.168.1.71",
##  user="bagus",
##  passwd="d3b5f9",
##  database="crab"
##)

    db = mysql.connector.connect(
        host="localhost",
        user="root",
        passwd="",
        database="crab"
    )

if db.is_connected():
    qry = db.cursor()
    print("Berhasil terhubung ke database")

def ambilgambar():
    #ALAMAT IP RASPBERRY PI
    response =
requests.get("http://192.168.88.50/larva/capture.php")
    t = time.localtime()
```

```

fn_img =
time.strftime("foto_larva_%d%m%y%H%M%S.jpg",t)

jsonRes = response.json()
d = eval(json.dumps(jsonRes))

path_tmp =
r"E:\02_DATA_KAMPUS\Thesis_nopi\Program\Larvae_Detectio
n\larva.jpg"

if os.path.exists(path_tmp):
    os.remove(path_tmp)

local_file = wget.download(d['url'])

tmp_img = Image.open('larva.jpg')
fn = ("E:/02_DATA_KAMPUS/tmp_larva/ori/",fn_img)
out_img = ".join(fn)
tmp_img.save(out_img)

sql = "insert into larva (kolam,asli,megalopa) values
(%s,%s,%s)"

val = ("1",fn_img,0)
qry.execute(sql,val)

db.commit()

print("ambil image berhasil")

def deteksi():

path = "E:/02_DATA_KAMPUS/tmp_larva/deteksi/"

sql = "select id,asli from larva where flag = 0"

```

```

qry.execute(sql)

res = qry.fetchall()

for x in res:

    a = x[0]

    b = x[1]

    c = ".join((path,b))

    print(c)

    commands = ["darknet.exe", "detector", "test",
"data/obj.data", "cfg/yolov_larva.cfg", "yolov_larva_last.weights"
, "-dont_show", c]

    os.system(' '.join(commands))

    output = subprocess.check_output(commands)

    output = output.decode("utf-8").split("\n")

    numM = len([i.split(":")[0] for i in output if i.split(":")[0] ==
'megalopa'])

    predicted_image = Image.open("predictions.jpg")

    t = time.localtime()

    fn_img =

time.strftime("foto_deteksi_%d%m%y%H%M%S.jpg",t)

    sms = time.strftime("%d-%m-%y %H:%M:%S",t)

    output_img = ".join((path,fn_img))

    predicted_image.save(output_img)

```

```

    sql = "update larva set deteksi = %s,megalopa = %s,flag =
1 where id = %s"

    val = (fn_img,numM,a)
    qry.execute(sql,val)

    m = ".join(("Pada ",fn_img,", ada ",str(numM)," megalopa
yang terdeteksi"))

    n = ".join(("Pada kolam 1, ada ",str(numM)," megalopa yang
terdeteksi pada ",sms))

    print(m)

    #Kondi

    if numM >= 0:

        sql = "insert into outbox
(DestinationNumber,TextDecoded,CreatorID)
values
(%s,%s,%s)"

        #NO HP PETANI/PENERIMA

        val = ("081334113678",n,"Novi")

        qry.execute(sql,val)

def looping():

while(True):

    try:

        t = time.localtime()

        ##current_time = time.strftime("%H:%M:%S",t)

```

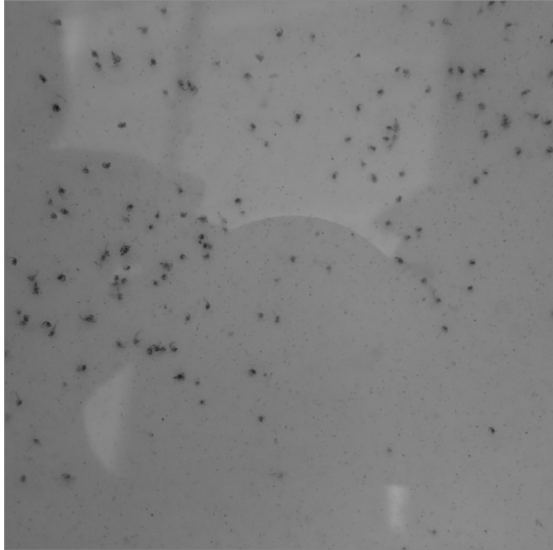
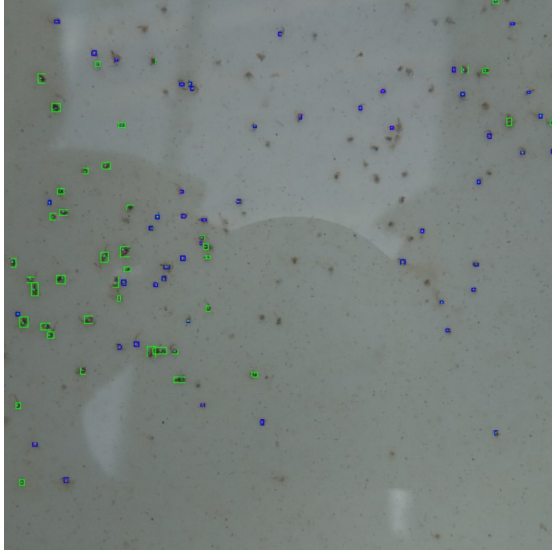
```

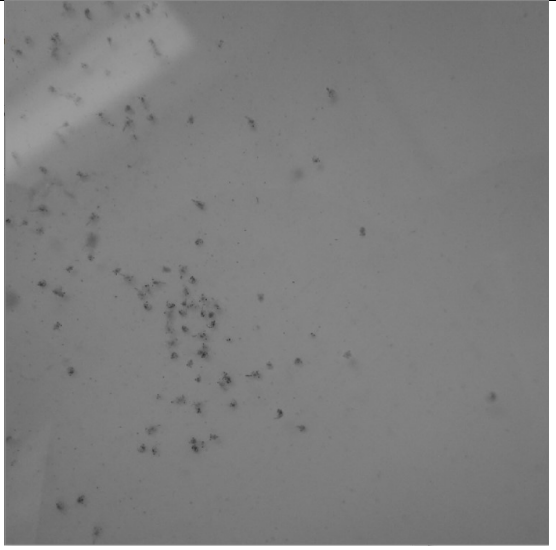
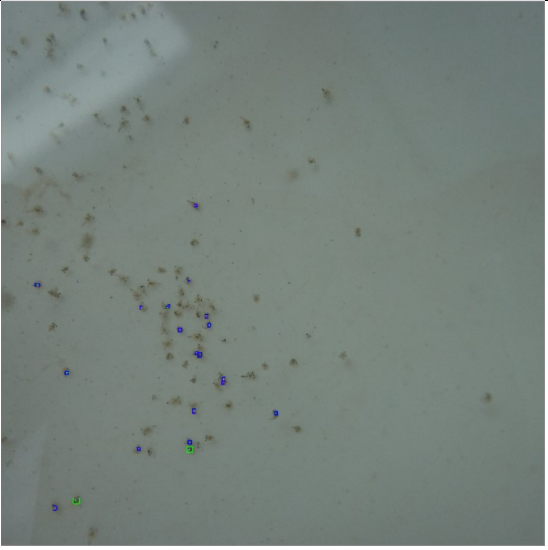
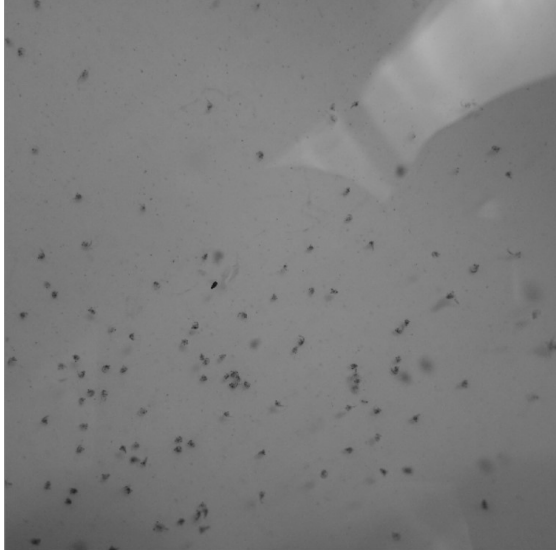
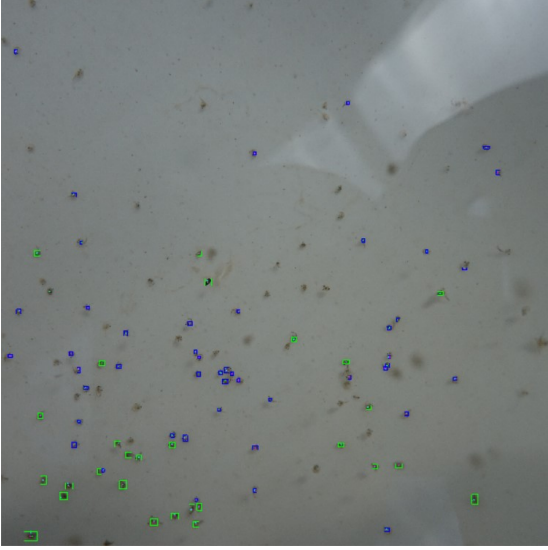
current_time =
time.strftime("foto_larva_%H%M%S.jpg",t)
print("Current Time = ", current_time)
ambilgambar()
deteksi()
time.sleep(10)
except KeyboardInterrupt:
print("Program distop oleh user")
break
looping()

```

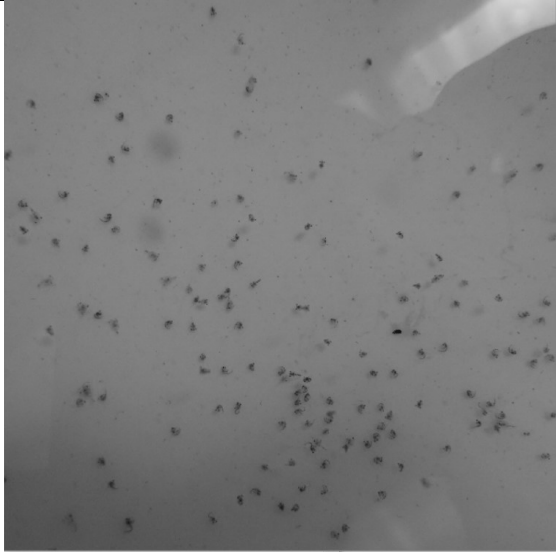
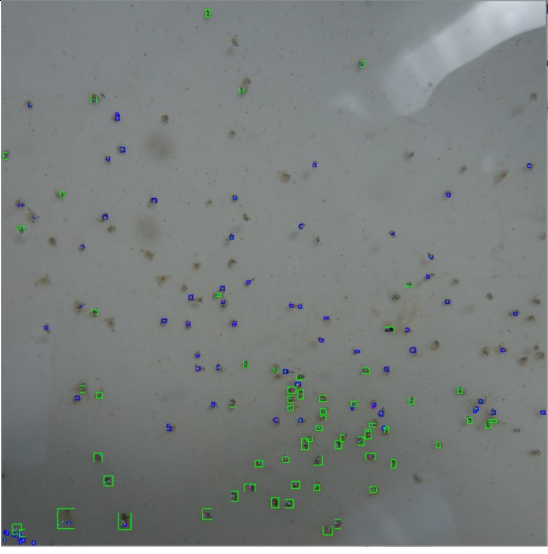
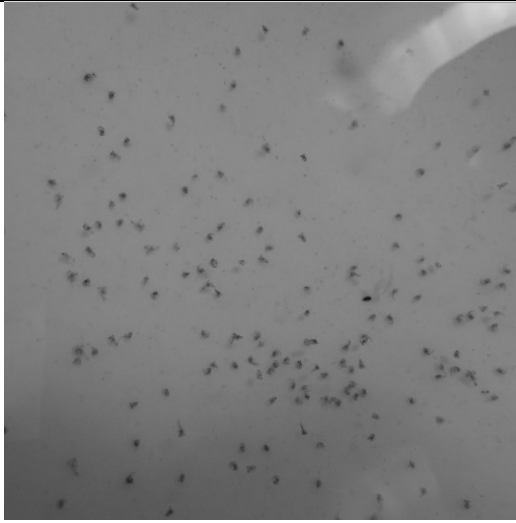
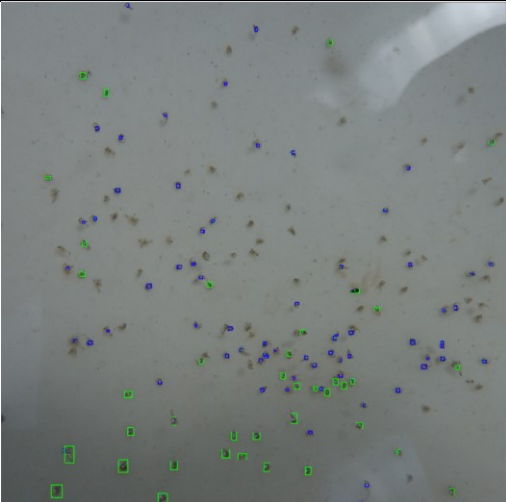
### 3. Gambar Hasil Pengujian Sistem

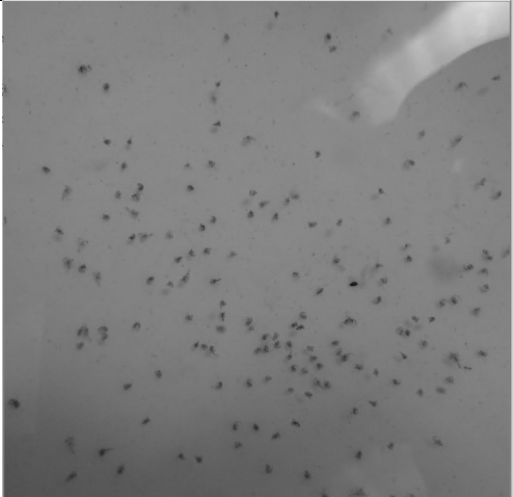
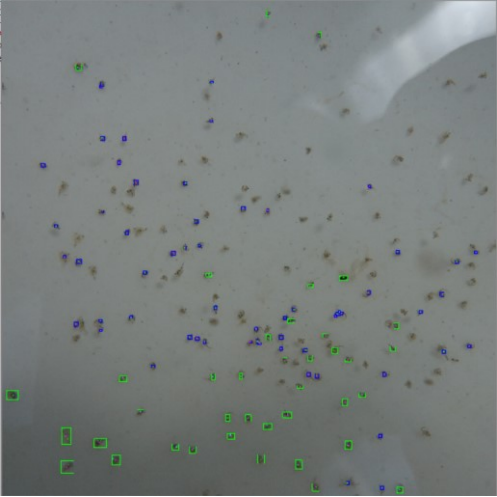

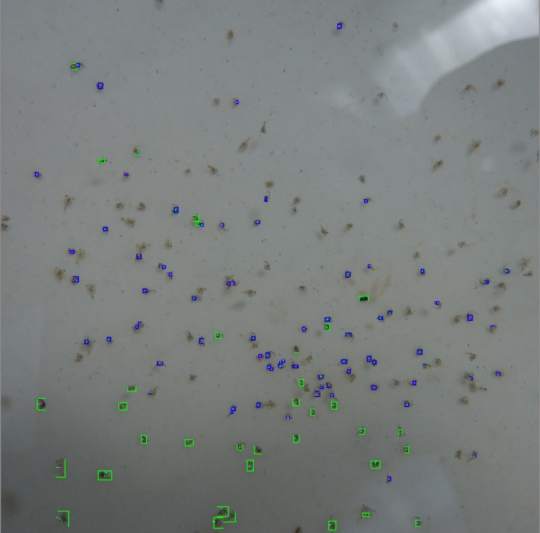
- a. Hasil pengujian sistem melalui tahapan preprocessing, morphological operasional dan klasifikasi SVM

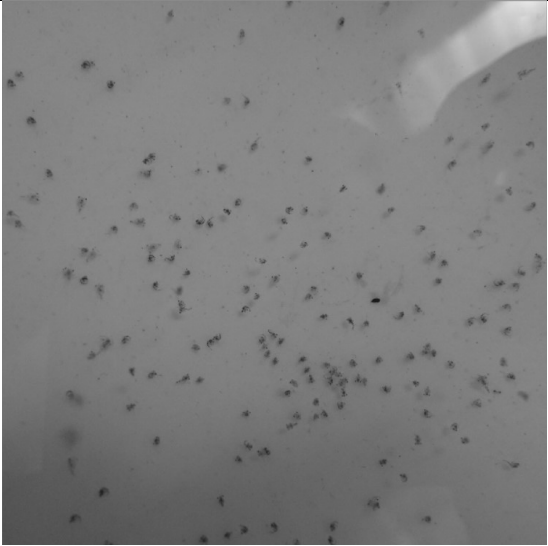
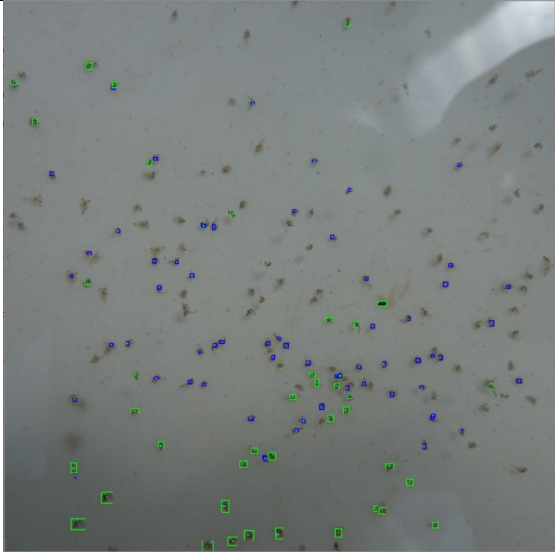
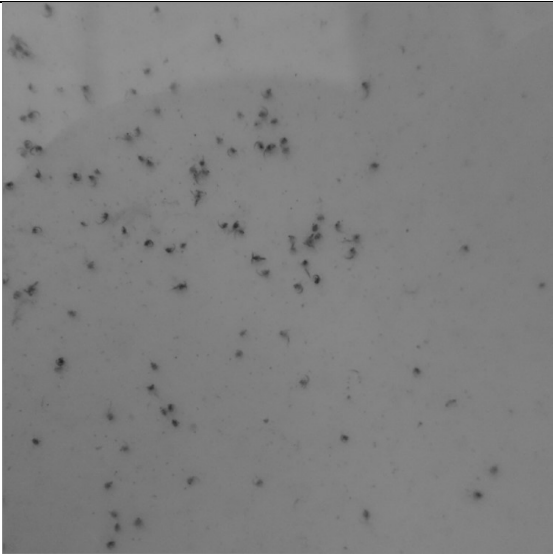
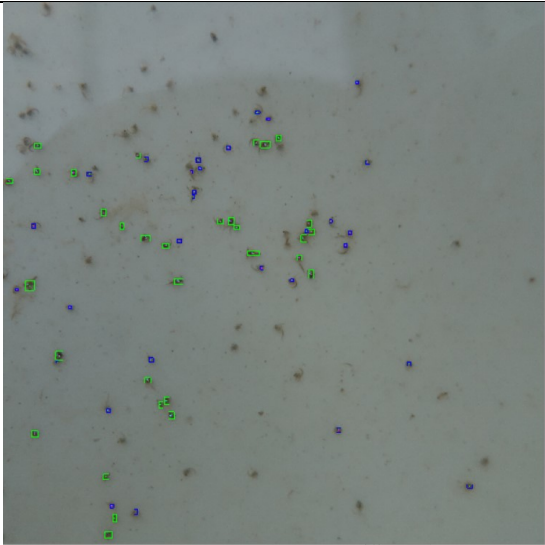
No	Gambar Uji	Gambar Hasil Deteksi
1		 Jumlah Megalopa : 41

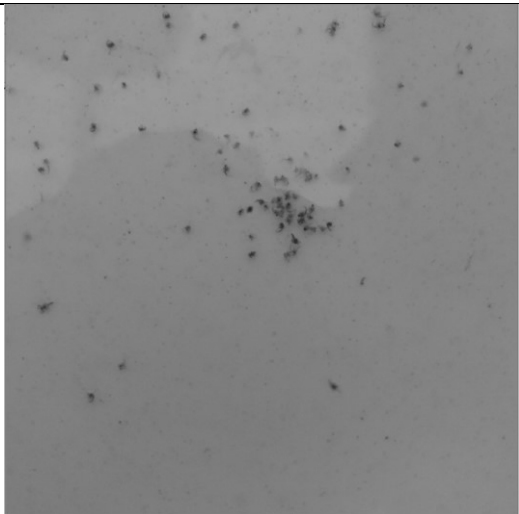
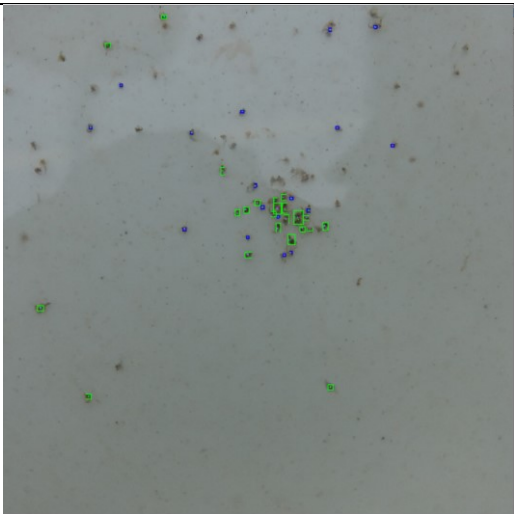
No	Gambar Uji	Gambar Hasil Deteksi
2		<p data-bbox="966 283 1193 315">Jumlah Zoea : 49</p>  <p data-bbox="966 892 1242 924">Jumlah Megalopa : 2</p> <p data-bbox="966 934 1193 966">Jumlah Zoea : 19</p>
3		 <p data-bbox="966 1554 1258 1585">Jumlah Megalopa : 29</p> <p data-bbox="966 1596 1193 1627">Jumlah Zoea : 47</p>





No	Gambar Uji	Gambar Hasil Deteksi
4		 <p data-bbox="966 840 1258 924">Jumlah Megalopa : 63 Jumlah Zoea : 72</p>
5		 <p data-bbox="966 1459 1258 1543">Jumlah Megalopa : 39 Jumlah Zoea : 68</p>


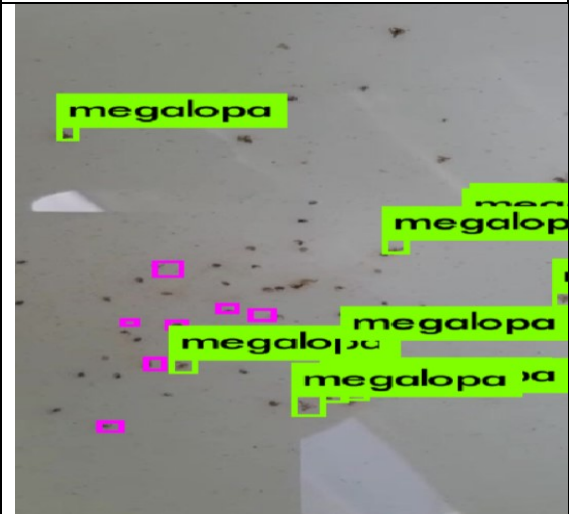
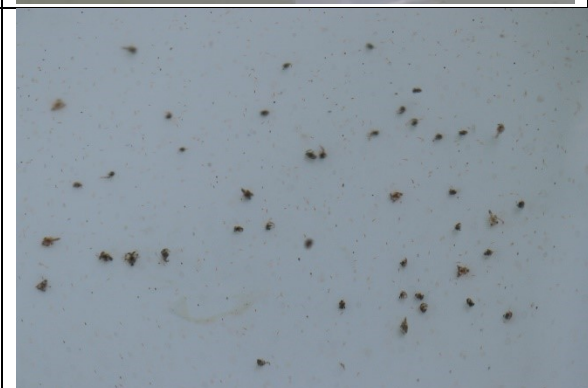
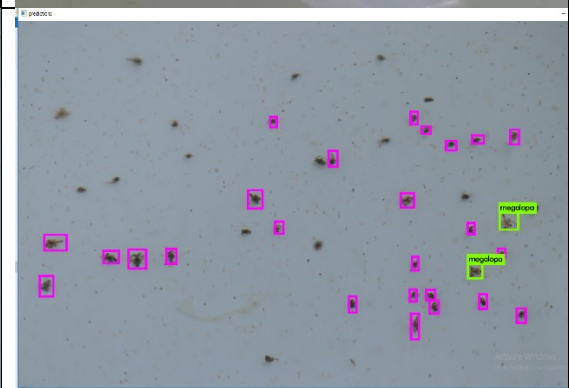

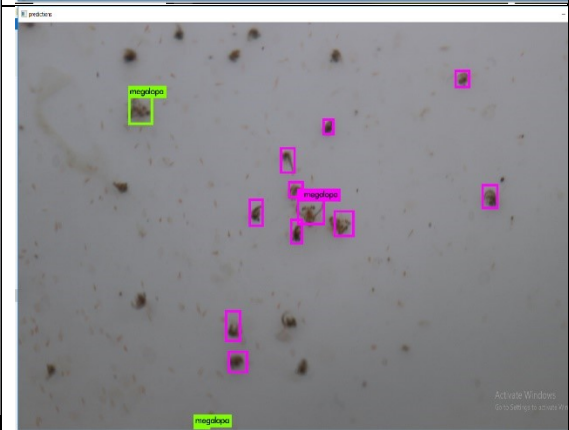
No	Gambar Uji	Gambar Hasil Deteksi
6		 <p data-bbox="967 789 1256 873">Jumlah Megalopa : 39 Jumlah Zoea : 59</p>
7		 <p data-bbox="967 1442 1256 1526">Jumlah Megalopa : 35 Jumlah Zoea : 75</p>

No	Gambar Uji	Gambar Hasil Deteksi
8		 <p data-bbox="964 846 1255 930">Jumlah Megalopa : 40 Jumlah Zoea : 62</p>
9		 <p data-bbox="964 1507 1255 1591">Jumlah Megalopa : 32 Jumlah Zoea : 30</p>



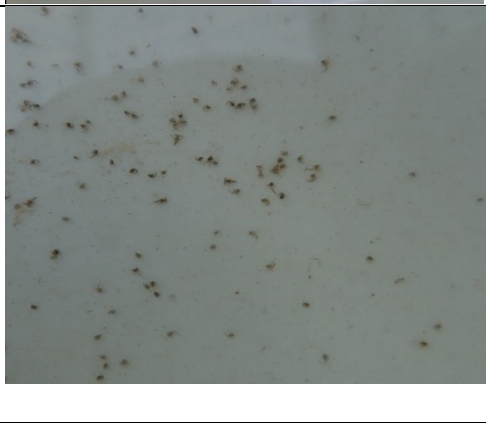
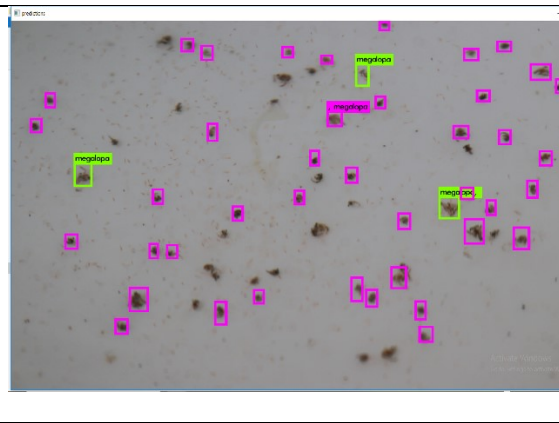

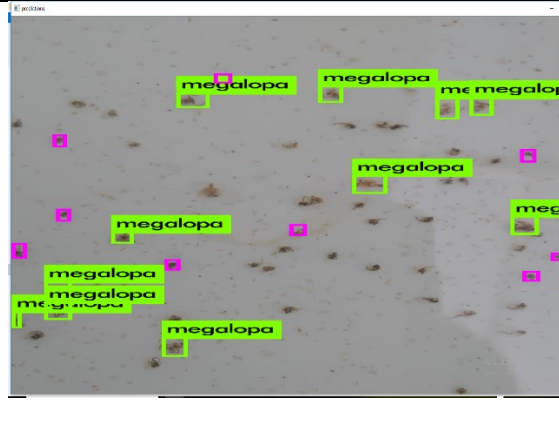
No	Gambar Uji	Gambar Hasil Deteksi
10		 <p>Jumlah Megalopa : 23 Jumlah Zoea : 17</p>


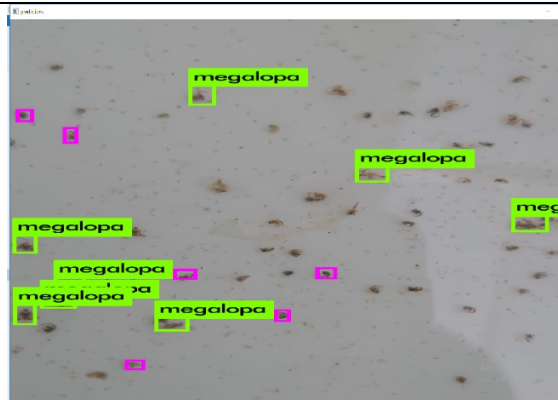

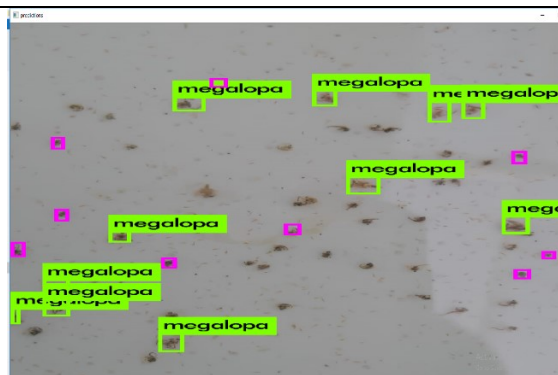
b. Hasil pengujian sistem menggunakan YOLOv3

No	Gambar Uji	Gambar Hasil Deteksi
1		 <p>Adara Winona 02/03/2023 10:00:00</p>

No	Gambar Uji	Gambar Hasil Deteksi
2		
3		
4		



No	Gambar Uji	Gambar Hasil Deteksi
5		
6		
7		

No	Gambar Uji	Gambar Hasil Deteksi
8		
9		
10	